Diseñando apps para móviles



Javier Cuello y José Vittone

Diseñando apps para móviles

Javier Cuello y José Vittone

Diseñando apps para móviles
©Javier Cuello
©José Vittone

Primera edición: junio de 2013
Versión 1.1
ISBN: 978-84-616-5070-5

Edición: Catalina Duque Giraldo
Ilustraciones: ©TugaMOVIL
Fotografías de los autores: ©Comitè Fotogràfic

Encuéntranos también en: **www.appdesignbook.com**

Tabla de contenidos

Presentación

Diseñar aplicaciones no es una tarea fácil, especialmente si no has trabajado antes con ellas o si vienes del mundo web, un contexto radicalmente diferente al que ahora nos enfrentamos.

Aunque parecen un fenómeno reciente, la realidad es que las aplicaciones hace tiempo que están entre nosotros. Últimamente, no solo se han vuelto más populares y atractivas para los usuarios, sino también para los diseñadores y desarrolladores que están sacando provecho de las posibilidades que ofrecen las nuevas pantallas de teléfono de mayor calidad. En una app este avance tecnológico se traduce en mejores experiencias, apoyadas en el diseño visual que ahora tiene más importancia y responsabilidad.

Para los diseñadores es todo un desafío empezar a diseñar para móviles, pero también, es una buena oportunidad para meterse en este ámbito donde los clientes demandan cada vez más y mejores herramientas de comunicación y promoción.

Por supuesto, tampoco nos hemos olvidado de los programadores, pues el mercado exige aplicaciones de calidad que puedan destacarse de las demás y no basta solo con que se vean bien. Deben, además, funcionar y ser fáciles de usar.

Tanto diseñadores como desarrolladores deberían entonces trabajar codo a codo, para lograr aplicaciones que cuiden todos y cada uno de los detalles. Después de todo, la primera impresión es la que cuenta.

Por eso, en los capítulos de este libro, vamos a desarrollar varios conceptos que te servirán para producir proyectos de calidad.

¿Por qué este libro?

El libro que tienes en tus manos surgió originalmente como nuestra respuesta al poco material en español sobre diseño de aplicaciones y fue traducido posteriormente al inglés. En cualquier caso —y en cualquier idioma— la documentación sobre este tema está dispersa por la red, fragmentada o es difícil de encontrar. Cuando nos estábamos iniciando en este trabajo, varias veces quisimos buscar algo y terminamos con una fea sensación de desconcierto al no encontrarlo: el ligero mareo mental de sentirse perdido en un inmenso mar de información.

Nuestros comienzos en el diseño de apps no fueron nada fáciles. Todo era muy diferente y no teníamos a nadie que nos explicara, ni un lugar donde despejar nuestras dudas. Ya han pasado unos pocos —pero valiosos— años desde entonces y lo que hemos aprendido, casi a la fuerza y sobre la marcha, nos ha servido mucho... ¡muchísimo!

Queremos compartir esto contigo para que no tengas que pasar años antes de aprenderlo. Aun así, te animamos a ser curioso y a investigar también por tu cuenta. Toma este libro como un punto de partida que te llevará por nuevos rumbos, desde donde podrás construir y recorrer tu propio camino.

Sobre todas las cosas: ¡haz!, ¡prueba!, ¡equivócate! Empieza a diseñar lo antes posible, aun cuando sean pequeños proyectos personales. Observa cómo lo hacen los demás. Si no tienes un teléfono, busca uno y explóralo. Investiga y juega con él hasta que sientas que ya lo conoces, que no tiene secretos para ti. También así aprenderás.

Cómo está estructurado

En los siguientes capítulos te guiaremos a través del proceso de diseño de una aplicación, desde la idea inicial hasta la publicación en las diferentes tiendas, pasando por el diseño de interacción y de interfaz. Y en cada uno de ellos encontrarás enlaces relacionados para que puedas expandir tus conocimientos sobre algunos conceptos o temas específicos.

Este libro no abarca solo el diseño visual. Quisimos ir un poco más allá, tocando temas de mercado, pruebas con usuarios o promoción de la app. Creemos firmemente en un diseñador que está presente en todo el proceso de desarrollo, que se involucra y compromete en cada una de las etapas para asegurar un producto de calidad.

Aunque hablamos del diseño de aplicaciones de manera global, en varias oportunidades establecemos paralelos entre los sistemas operativos para móviles más comunes: Android, iOS y Windows Phone. Claramente, hemos elegido las dos primeras plataformas porque hoy en día son las más importantes; la última, si bien no es tan significativa en cantidad de usuarios —todavía— ha roto un poco los esquemas a nivel de diseño presentando una propuesta que se separa de las demás.

Adicionalmente, también descubrirás entrevistas con algunos profesionales reconocidos del mundo del diseño, como Loren Brichter, Dustin Mierau, Irene Pereyra o Erik Spiekermann, entre otros. Los hemos contactado especialmente para que nos cuenten su visión sobre el mundo de las apps, enriqueciendo enormemente los contenidos.

Al final del libro encontrarás un glosario con términos comunes de este entorno, en caso de que necesites familiarizarte con ellos. Siéntete libre de consultarlo cada vez que no entiendas algo, queremos asegurarnos de que comprendas el significado de lo que te estamos contando.

Acerca de los autores



Javier «Simón» Cuello

Nació en Mendoza, Argentina. Una pequeña ciudad al oeste del país y al pie de las montañas, conocida por sus árboles, acequias y la calidad de sus vinos. «Simón» tuvo la suerte de querer ser diseñador gráfico desde muy pequeño y, a medida que fue creciendo, a su formación visual le fue sumando un particular interés por lo interactivo y las nuevas tecnologías.

Sus últimos años en Barcelona son simplemente la confirmación de esto. Ha trabajado en diseño de aplicaciones móviles desde las primeras versiones de iOS y Android, para clientes como Yahoo! y Zara.

Además de diseñar, a «Simón» le encanta viajar y trabajar en equipos pequeños. Mientras no esté cumpliendo su sueño de recorrer el mundo y diseñar al mismo tiempo, lo encontrarás pensando ideas para algún proyecto independiente.

Si quieres conocer más a «Simón», deberías seguirlo en Twitter: **@millonestarde**.



José Vittone

José es también originario de Argentina, pero del interior de la provincia de Buenos Aires. Actualmente vive en Barcelona, donde hace unos cuatro años conoció a «Simón» cuando eran compañeros de Máster en Elisava.

Trabajó en Usolab y allí aprendió lo que en realidad significa la usabilidad y el diseño centrado en el usuario. Desde hace ocho años diseña interfaces, participando tanto en la conceptualización como en la producción de los proyectos. Durante el último tiempo, ha estado haciendo malabares con varias aplicaciones móviles para múltiples plataformas.

Le apasiona el cuidado de los pequeños detalles que hacen grandes diferencias, tiene una curiosidad innata para investigar cómo funcionan las cosas y siempre está al día con las nuevas tecnologías.

José publica regularmente en Twitter, no te pierdas sus comentarios y síguelo en: @josevittone.

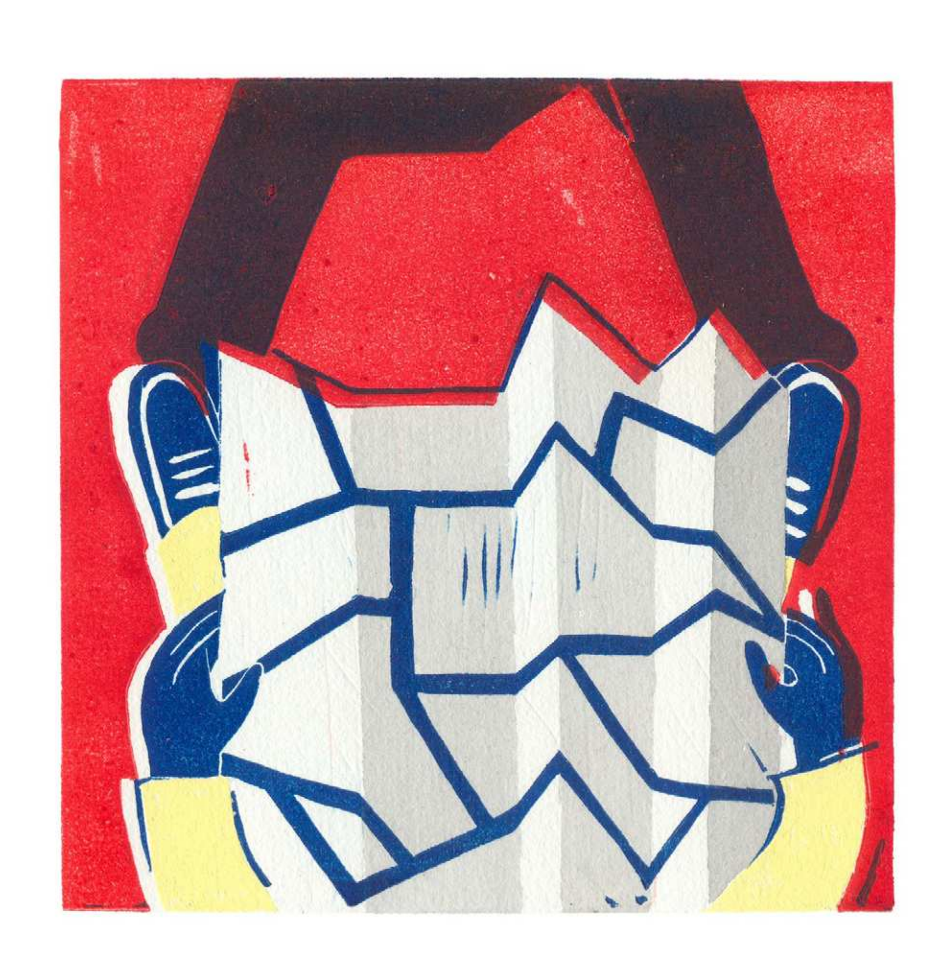
Acerca de las ilustraciones



En cada capítulo vas a encontrar una presentación de los contenidos acompañada de una imagen. Estos dibujos son obra de TugaMOVIL[1], un pequeño estudio polaco-chileno que trabaja desde Barcelona y a quienes tenemos la fortuna de contar entre nuestros amigos cercanos. Presta mucha atención a estas ilustraciones creadas por Maga y Seba que te acompañarán en el viaje que estamos a punto de comenzar.

¡Ahí vamos!

1 http://www.tugamovil.com/appdesignbook/



1

Las aplicaciones

Aunque no parezca, las aplicaciones llevan tiempo entre nosotros. Antes de empezar a diseñarlas, queremos que las conozcas: ¿Cuáles son sus tipos y características? ¿Qué diferencias hay entre ellas y con una web móvil?

¿QUÉ SON LAS APLICACIONES?



FIG 1.1.
En la AppStore hay cientos de miles de apps disponibles.

Las aplicaciones —también llamadas apps— están presentes en los teléfonos desde hace tiempo; de hecho, ya estaban incluidas en los sistemas operativos de Nokia o Blackberry años atrás. Los móviles de esa época, contaban con pantallas reducidas y muchas veces no táctiles, y son los que ahora llamamos feature phones, en contraposición a los *smartphones*, más actuales.

En esencia, una aplicación no deja de ser un software. Para entender un poco mejor el concepto, podemos decir que las aplicaciones son para los móviles lo que los programas son para los ordenadores de escritorio.

Actualmente encontramos aplicaciones de todo tipo, forma y color, pero en los primeros teléfonos, estaban enfocadas en mejorar la productividad personal: se trataba de alarmas, calendarios, calculadoras y clientes de correo.

Hubo un cambio grande con el ingreso de iPhone al mercado, ya que con él se generaron nuevos modelos de negocio que hicieron de las aplicaciones algo rentable, tanto para desarrolladores como para los mercados de aplicaciones, como App Store, Google Play y Windows Phone Store.

Al mismo tiempo, también mejoraron las herramientas de las que disponían diseñadores y programadores para desarrollar apps, facilitando la tarea de producir una aplicación y lanzarla al mercado, incluso por cuenta propia.

DIFERENCIAS ENTRE APLICACIONES Y WEB MÓVILES

Las aplicaciones comparten la pantalla del teléfono con las webs móviles, pero mientras las primeras tienen que ser descargadas e instaladas antes de usar, a una web puede accederse simplemente usando Internet y un navegador; sin embargo, no todas pueden verse correctamente desde una pantalla generalmente más pequeña que la de un ordenador de escritorio.

Las que se adaptan especialmente a un dispositivo móvil se llaman «web responsivas» y son ejemplo del diseño líquido, ya que se puede pensar en ellas como un contenido que toma la forma del contenedor, mostrando la información según sea necesario. Así, columnas enteras, bloques de texto y gráficos

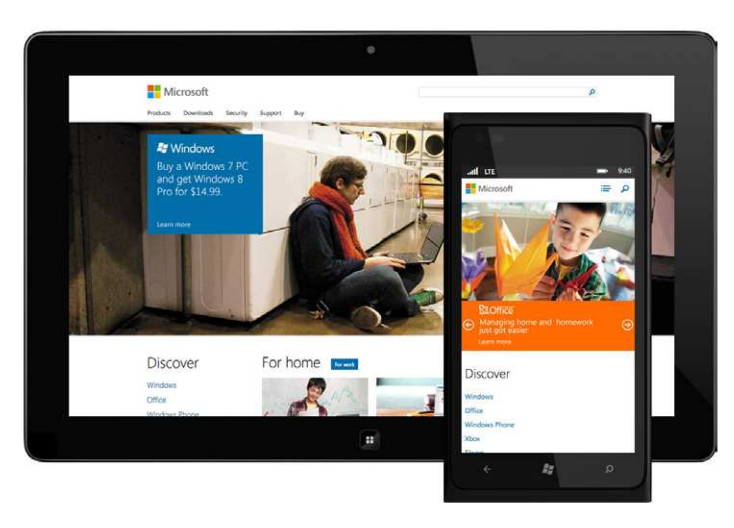


FIG 1.2. *El diseño «responsivo» se adapta dependiendo del dispositivo donde es visualizado.*

de una web, pueden acomodarse en el espacio de una manera diferente —o incluso desaparecer— de acuerdo a si se entra desde un teléfono, una tableta o un ordenador.

Quienes cuentan ya con una «web responsiva» pueden plantearse la necesidad de diseñar una aplicación, pero la respuesta a si esto es o no necesario, depende de entender tanto los objetivos de negocio, como las características que diferencian las aplicaciones de las webs.

Por ejemplo, las aplicaciones pueden verse aun cuando se está sin conexión a Internet, además, pueden acceder a ciertas características de hardware del teléfono —como los sensores—, capacidades que actualmente están fuera del alcance de las webs. Por lo anterior, puede decirse que una aplicación ofrece una mejor experiencia de uso, evitando tiempos de espera excesivos y logrando una navegación más fluida entre los contenidos.

No siempre hay que elegir entre una u otra. Webs y aplicaciones no son competidoras, más bien, pueden complementarse entre ellas; por ejemplo, una web puede ser útil como canal de información para motivar la descarga de la aplicación.

Primero el móvil

Es posible que cuando llegue la hora de diseñar una aplicación ya exista una web como antecedente. En esos casos, la app tiene que tomar las funciones y contenidos que se han pensando para la web y adaptarlos para que tengan sentido, de acuerdo al tamaño de pantalla y a la forma de interacción de un móvil.

En otros casos, el diseño comienza desde cero, cuando todavía no hay ni web ni aplicación, y hay que decidirse por cual de ellas empezar. Aquí es donde adquiere más trascendencia el concepto de *mobile first*, que implica plantear el proceso de diseño teniendo en cuenta el móvil en primer lugar.

La ventaja de esta forma de trabajar es que el pensar en el móvil como punto de partida, obliga a concentrarse en lo esencial de un producto y a hacer foco solo en lo que tiene sentido para este dispositivo.

Una vez que la aplicación está diseñada, puede preguntarse cuál es la mejor forma de llevar lo hecho para el teléfono a una pantalla de ordenador o a otros dispositivos, extendiendo y escalando el contenido y repensando la diagramación. Todos los dispositivos tienen usos diferentes, y en el momento de adaptar el diseño, hay que tener en cuenta las características particulares de cada uno de ellos.

Mobile first es una propuesta de trabajo que ha surgido recientemente; una tendencia emergente que aún está por consolidarse. Actualmente, es solo una manera de afrontar el proceso de diseño y como tal, puede evaluarse la comodidad que se tiene trabajando de esta forma antes de empezar[1].

El proceso de diseño y desarrollo de una app

El proceso de diseño y desarrollo de una aplicación, abarca desde la concepción de la idea hasta el análisis posterior a su publicación en las tiendas. Durante las diferentes etapas, diseñadores y desarrolladores trabajan —la mayor parte del tiempo— de manera simultánea y coordinada.

Hemos resumido las fases de este proceso solo desde la perspectiva del diseño y desarrollo, es decir, sin tener en cuenta los roles de coordinación, la participación del cliente, ni los accionistas de la empresa.

Cada una de las etapas —excepto la de desarrollo— se explica más extensamente en los capítulos siguientes, detallando procesos y metodologías para ir avanzando entre ellas.

1 Wroblewski, Luke. Mobile First. http://www.lukew.com/ff/entry.asp?933

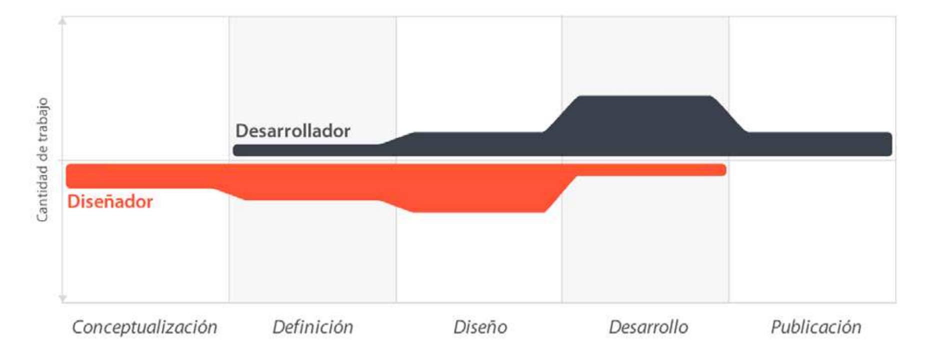


FIG 1.3.
El proceso de diseño abarca diferentes etapas donde diseñador y desarrollador trabajan simultáneamente.

1. Conceptualización

El resultado de esta etapa es una idea de aplicación, que tiene en cuenta las necesidades y problemas de los usuarios. La idea responde a una investigación preliminar y a la posterior comprobación de la viabilidad del concepto.

- Ideación
- Investigación
- Formalización de la idea

2. Definición

En este paso del proceso se describe con detalle a los usuarios para quienes se diseñará la aplicación, usando metodologías como «Personas» y «Viaje del usuario». También aquí se sientan las bases de la funcionalidad, lo cual determinará el alcance del proyecto y la complejidad de diseño y programación de la app.

- Definición de usuarios
- Definición funcional

3. Diseño

En la etapa de diseño se llevan a un plano tangible los conceptos y definiciones anteriores, primero en forma de *wireframes*, que permiten crear los primeros prototipos para ser probados con usuarios, y posteriormente, en un diseño visual acabado que será provisto al desarrollador, en forma de archivos separados y pantallas modelo, para la programación del código.

- *Wireframes*
- Prototipos
- Test con usuarios
- Diseño visual

4. Desarrollo

El programador se encarga de dar vida a los diseños y crear la estructura sobre la cual se apoyará el funcionamiento de la aplicación. Una vez que existe la versión inicial, dedica gran parte del tiempo a corregir errores funcionales para asegurar el correcto desempeño de la app y la prepara para su aprobación en las tiendas.

- Programación del código
- Corrección de *bugs*

5. Publicación

La aplicación es finalmente puesta a disposición de los usuarios en las tiendas. Luego de este paso trascendental se realiza un seguimiento a través de analíticas, estadísticas y comentarios de usuarios, para evaluar el comportamiento y desempeño

de la app, corregir errores, realizar mejoras y actualizarla en futuras versiones.

- Lanzamiento
- Seguimiento
- Actualización

Tipos de aplicaciones según su desarrollo

A nivel de programación, existen varias formas de desarrollar una aplicación. Cada una de ellas tiene diferentes características y limitaciones, especialmente desde el punto de vista técnico.

Aunque a primera vista esto no parezca incumbencia del diseñador, la realidad es que el tipo de aplicación que se elija, condicionará el diseño visual y la interacción.



Fig 1.4. Las aplicaciones nativas permiten aprovechar el sistema de notificaciones.

Aplicaciones nativas

Las aplicaciones nativas son aquellas que han sido desarrolladas con el software que ofrece cada sistema operativo a los programadores, llamado genéricamente *Software Development Kit* o SDK. Así, Android, iOS y Windows Phone tienen uno diferente y las aplicaciones nativas se diseñan y programan específicamente para cada plataforma, en el lenguaje utilizado por el SDK.

Este tipo de apps se descarga e instala desde las tiendas de aplicaciones —con ciertas excepciones en el caso de Android, que veremos en el capítulo «Lanzando la app»— sacando buen partido de las diferentes herramientas de promoción y marketing de cada una de ellas.

Las aplicaciones nativas se actualizan frecuentemente y en esos casos, el usuario debe volver a descargarlas para obtener la última versión, que a veces corrige errores o añade mejoras.

Una característica generalmente menospreciada de las apps nativas, es que pueden hacer uso de las notificaciones del sistema operativo para mostrar avisos importantes al usuario, aun cuando no se esté usando la aplicación, como los mensajes de Whatsapp, por ejemplo.

Además, no requieren Internet para funcionar, por lo que ofrecen una experiencia de uso más fluida y están realmente integradas al teléfono, lo cual les permite utilizar todas las características de hardware del terminal, como la cámara y los sensores (GPS, acelerómetro, giróscopo, entre otros).

A nivel de diseño, esta clase de aplicaciones tiene una interfaz basada en las guías de cada sistema operativo, logrando mayor coherencia y consistencia con el resto de aplicaciones y con el propio SO. Esto favorece la usabilidad y beneficia directamente al usuario que encuentra interfaces familiares.

Aplicaciones web

La base de programación de las aplicaciones web —también llamadas *webapps*— es el HTML, conjuntamente con JavaScript y CSS, herramientas ya conocidas para los programadores web.

En este caso no se emplea un SDK, lo cual permite programar de forma independiente al sistema operativo en el cual se usará la aplicación. Por eso, estas aplicaciones pueden ser fácilmente utilizadas en diferentes plataformas sin mayores inconvenientes y sin necesidad de desarrollar un código diferente para cada caso particular.





FIG 1.5. Facebook cuenta tanto con una webapp como con una app nativa.

Las aplicaciones web no necesitan instalarse, ya que se visualizan usando el navegador del teléfono como un sitio web normal. Por esta misma razón, no se distribuyen en una tienda de aplicaciones, sino que se comercializan y promocionan de forma independiente.

Al tratarse de aplicaciones que funcionan sobre la web, no es necesario que el usuario reciba actualizaciones, ya que siempre va a estar viendo la última versión. Pero, a diferencia de las apps nativas, requieren de una conexión a Internet para funcionar correctamente.

Adicionalmente, tienen algunas restricciones e inconvenientes en factores importantes como gestión de memoria y no permiten aprovechar al máximo la potencia de los diferentes componentes de hardware del teléfono.

Las aplicaciones web suelen tener una interfaz más genérica e independiente de la apariencia del sistema operativo, por lo que la experiencia de identificación del usuario con los elementos de navegación e interacción, suele ser menor que en el caso de las nativas.

Aplicaciones híbridas

Este tipo de aplicaciones es una especie de combinación entre las dos anteriores. La forma de desarrollarlas es parecida a la de una aplicación web —usando HTML, CSS y JavaScript—, y una vez que la aplicación está terminada, se compila o empaqueta de forma tal, que el resultado final es como si se tratara de una aplicación nativa.

Esto permite casi con un mismo código obtener diferentes aplicaciones, por ejemplo, para Android y iOS, y distribuirlas en cada una de sus tiendas.



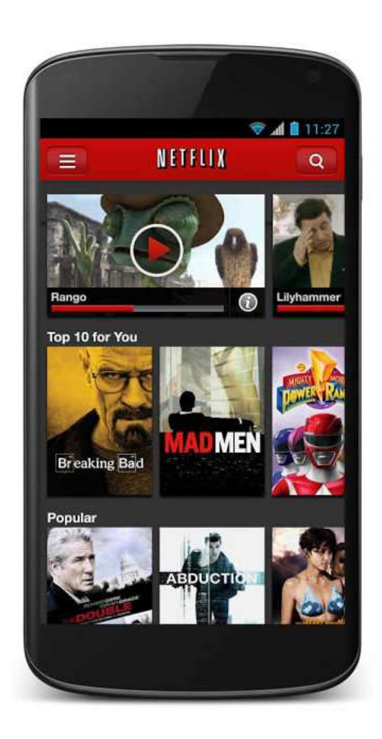


FIG 1.6. Netflix tiene una aplicación híbrida que se ve prácticamente igual en iOS y en Android.

A diferencia de las aplicaciones web, estas permiten acceder, usando librerías, a las capacidades del teléfono, tal como lo haría una app nativa[2].

Las aplicaciones híbridas, también tienen un diseño visual que no se identifica en gran medida con el del sistema operativo. Sin embargo, hay formas de usar controles y botones nativos de cada plataforma para apegarse más a la estética propia de cada una.

Existen algunas herramientas para desarrollar este tipo de aplicaciones. Apache Cordova[3] es una de las más populares, pero hay otras, como Icenium[4], que tienen la misma finalidad.

2 SEVEN, Doug. What is a Hybrid Mobile App? http://icenium.com/community/blog/icenium-team-blog/2012/06/14/what-is-a-hybrid-mobile-app-
3 http://cordova.apache.org/
4 http://www.icenium.com/

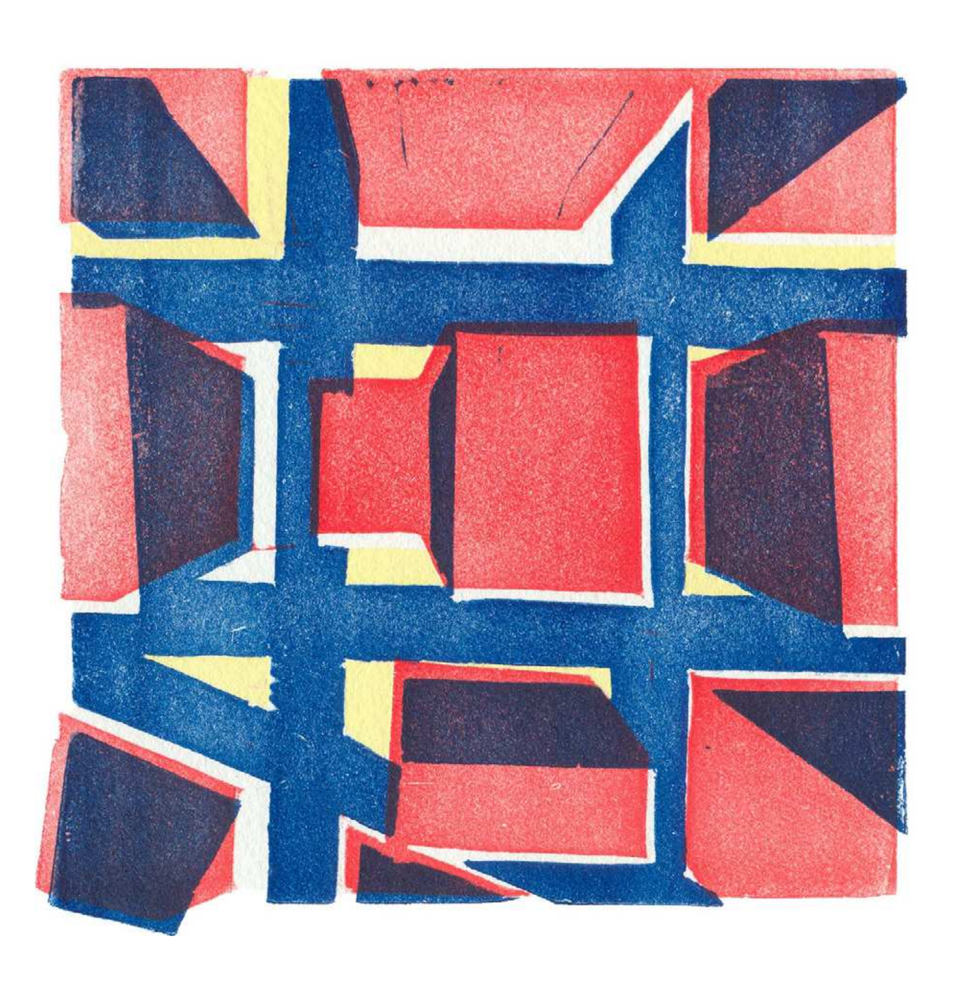
¿Cuál deberías usar?

Dadas las características de cada una de las aplicaciones, decidirse por una u otra estará determinado por unos pocos factores fundamentales y por la forma en que afectan finalmente la experiencia de uso. Cuando la disponibilidad de la app sin Internet, la posibilidad de usar notificaciones y el acceso a los recursos de hardware del teléfono sean importantes, una aplicación nativa será la opción más indicada.

Si ninguna de estas cosas es realmente importante para la aplicación, quizás sea más fácil diseñar una aplicación web, si es que ya se dispone del conocimiento para ello, heredado del desarrollo de sitios web. En este caso, el costo de desarrollo es más bajo y la forma de trabajar un poco más ágil.

Independientemente de esto, las aplicaciones nativas son las que ofrecen una mejor experiencia de uso y sobre todo, rendimiento. Algunas apps como Facebook o LinkedIn, que antes eran híbridas, han pasado a ser nativas por este motivo. Adicionalmente, ellas responden más a las guías de diseño de cada sistema operativo.

Por lo anterior, nos enfocaremos en las aplicaciones nativas. De aquí en adelante, y por el resto de los capítulos que quedan por venir, vamos a estudiarlas, conocerlas y a ver qué las hace diferentes.



2

Entendiendo las posibilidades

Un poco antes de desarrollar la idea de una aplicación hay que tomar una serie de decisiones que marcarán el rumbo del proyecto. En este capítulo te mostramos las diferentes opciones para sacar rédito económico de tu aplicación y el equipo y recursos que necesitarás para seguir adelante.

Categorías de aplicaciones

Una forma de agrupar las aplicaciones es de acuerdo al tipo de contenido que ofrecen al usuario. La categoría a la que se pertenezca condicionará, a nivel de diseño, con qué nivel de detalle contará la interfaz y también influirá en las posibilidades de monetización de la aplicación.

Puede parecer injusto etiquetar las apps, porque difícilmente suelen pertenecer a una sola categoría. Incluso, muchas veces pueden estar en más de una por igual, pero siendo algo rigurosos, las hemos dividido de la siguiente forma:

Entretenimiento

Este es el lugar donde viven las apps de juegos y aquellas que de una forma u otra, proponen diversión para el usuario. Gráficos, animaciones y efectos de sonido intentan mantener la atención constante e ininterrumpida en lo que está sucediendo en la pantalla.

Generalmente, presentan un diseño no tan apegado a las pautas de su plataforma. Un ejemplo es Angry Birds[1], que propone gráficos similares entre los diferentes sistemas operativos.



Fig 2.1. Angry Birds es uno de los juegos para móviles más populares del momento.

1 http://www.angrybirds.com/

En cuanto al modelo de negocio, son flexibles porque pueden descargarse pagando por versiones completas u ofrecer otras posibilidades de compra: por ítems, por niveles, etc.

Sociales

Las aplicaciones sociales son aquellas que se orientan principalmente a la comunicación entre personas, construcción de redes de contactos e interacción entre usuarios. Es por todos conocido el caso de Facebook, pero aquí también se encuentran otras como Path[2], Twitter[3] e Instagram[4].

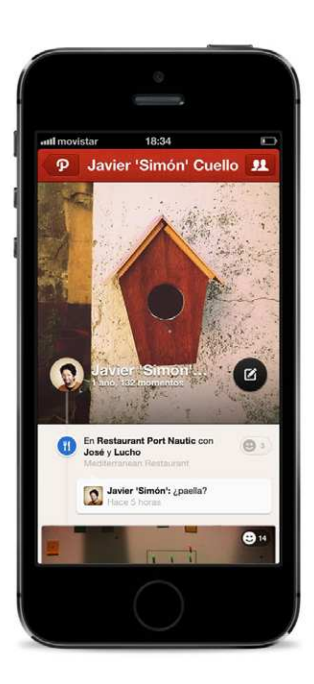
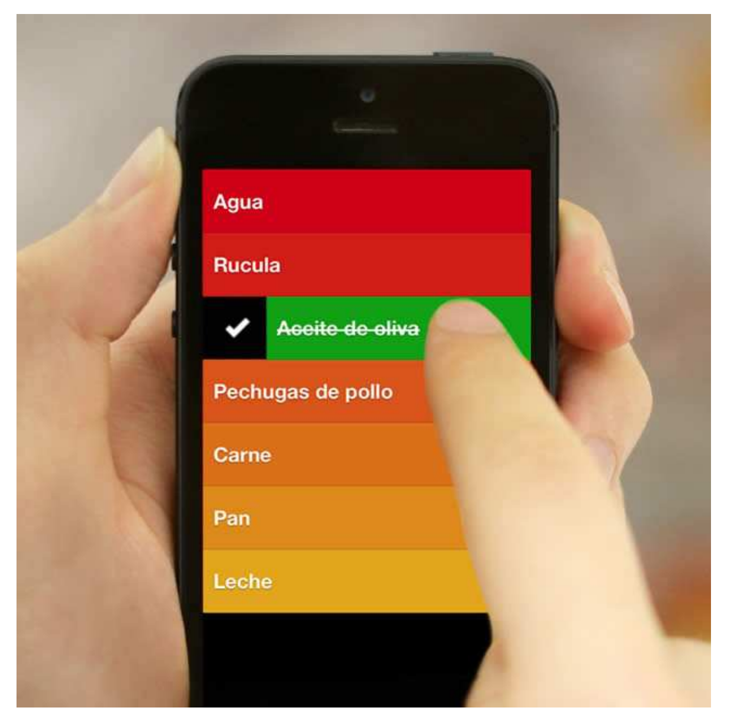




FIG 2.2. *Path es una red social que permite 150 amigos como máximo.*

2 https://path.com/
3 https://twitter.com/
4 http://instagram.com/

FIG 2.3. Clear revolucionó las apps de tareas con su estilo particular.



Suelen ser gratuitas y su modelo de negocio radica en la información personal que se obtiene de los usuarios o en las compras dentro de la app, por ejemplo, los stickers en el caso de Path.

Utilitarias y productividad

Más asociadas con el sector empresarial, las aplicaciones utilitarias proporcionan herramientas para solucionar problemas bastante específicos y se basan en la ejecución de tareas concretas, cortas y rápidas. En este caso se privilegia la eficiencia sobre todo lo demás.

En este apartado comúnmente encontramos las listas de tareas (Clear[5] y Flow[6]) o aquellas apps orientadas a equipos de trabajo (Basecamp[7] y Evernote[8]), donde los usuarios dan más valor a aquellas herramientas que permitan simplificar sus tareas diarias.

En esta categoría el modelo de negocio es variable. Si la aplicación se encuentra solo disponible para móvil, lo normal es que se pague por la descarga, como en el caso de Clear. En cambio, las que están asociadas a un servicio en la nube por el cual ya se paga, como Basecamp, se descargan de forma gratuita.



FIG 2.4. Articles es una buena alternativa a la app de Wikipedia, aunque es de pago.

5 http://www.realmacsoftware.com/clear
6 http://www.getflow.com/
7 http://basecamp.com/
8 https://evernote.com/

Educativas e informativas

Las aplicaciones educativas y de información se usan como transmisores de conocimiento y noticias. En estas apps se privilegia el acceso al contenido, por este motivo, la legibilidad, facilidad de navegación y herramientas de búsqueda son fundamentales.

Algunas de ellas, como Articles[9], se pagan; mientras que otras, como Wikipedia, son gratuitas.

Creación

Estas aplicaciones ponen el foco en la creatividad del usuario y en ofrecerle herramientas para potenciarla. Por ejemplo, aquellas que permiten editar vídeos, retocar fotografías, producir sonidos o escribir.

Aunque suelen ser de pago, algunas ofrecen versiones gratuitas —no tan completas— o añaden componentes extra que deben pagarse individualmente. Un ejemplo de este tipo de aplicaciones es Paper[10], que además de las herramientas básicas, también ofrece elementos de dibujo adicionales para comprar dentro de la misma app.

Aplicaciones gratis, de pago... o un poco de las dos

Sobre este tema hay y habrá mucho debate. Las apps son un tipo de producto relativamente nuevo y, si bien muchas veces responden a modelos de negocio ya usados para otro tipo de software, en algunos casos hay incertidumbre sobre cuál es

9 http://sophiestication.com/articles/
10 http://www.fiftythree.com/paper



FIG 2.5. Paper para iPad es una app de creación que se basa en la metáfora del bloc de dibujo.

la mejor forma de obtener dinero con ellas o cuál puede ser la ganancia prevista.

Aún así, puede ser que el objetivo de la app no sea conseguir dinero, sino servir como canal de comunicación con los usuarios o como una forma de extender el alcance de marca. De cualquier forma, se trata de una decisión personal —en el caso de los desarrolladores independientes— o que responde a objetivos globales —para las marcas—.

Claramente, cada uno de los modelos —gratuito o de pago— tiene pros y contras. Aquí los explicamos para ayudarte a tomar una decisión más informada, aun cuando las dos alternativas se mezclan entre sí.

Apps gratuitas

Sin dudas el mayor beneficio que se puede obtener de una aplicación gratuita es el alcance o la cantidad de usuarios potenciales a los que puede llegar, ya que no hay ninguna barrera de entrada para que un usuario descargue la aplicación y empiece a probarla.

Este primer paso es fundamental para el conocimiento de la aplicación: quien la descargue no tiene nada que perder. Esto disminuye en cierta forma las expectativas sobre el producto —nadie espera que algo gratis sea necesariamente genial— y puede servir para ciertos casos en los que el producto aún tiene camino de desarrollo por delante hasta su maduración. En este caso, su colocación gratuita en el mercado, permite usar ciertos indicadores del comportamiento de los usuarios —cómo usan la app o qué tan frecuentemente, por ejemplo— para mejorar futuras versiones.

Un inconveniente a tener en cuenta con las aplicaciones gratuitas es que, debido a la gran competencia, es más difícil obtener visibilidad en los rankings de mejores aplicaciones de las diferentes tiendas. Además, requieren una cantidad total de descargas mayor que la necesaria en las aplicaciones de pago para llegar a los primeros puestos.

Una aplicación gratuita también puede ser servir como ventana para promocionar una versión de pago de ella misma u otras aplicaciones del mismo desarrollador.

Finalmente, que una app sea gratuita, no quiere decir que no se pueda obtener ningún dinero con ella, como se verá más adelante.

Apps de pago

Las aplicaciones de pago tienen las cosas más difíciles a la hora de ser exitosas porque requieren un gran número de descargas para ser rentables; además, poner un precio a la descarga suele ser una barrera difícil de sortear para el usuario que no se quiere arriesgar a pagar por algo que aún no conoce. Salvo casos puntuales, como Cut the rope[11], muchas de ellas no consiguen tener un alcance masivo.

11 http://www.cuttherope.ie/





FIG 2.6.
Cute the Rope y Whatsapp son dos apps que a pesar de ser de pago han conseguido muchísimas descargas.

Que un usuario esté dispuesto a pagar o no por una app, depende de varias cosas. Una de ellas tiene que ver con las alternativas gratuitas que pueda encontrar. Si dos aplicaciones son similares —a nivel de funcionalidad y diseño—, pero una de ellas es de pago y la otra no, obviamente es más probable que se descargue la aplicación gratuita.

La tienda donde se encuentre la app también determina las posibilidades de cobrar por ella. Por ejemplo, un usuario de iPhone está más acostumbrado a pagar por una aplicación que alguien que usa Android o Windows Phone; pero, quizás por esta misma razón, demanda más calidad para colmar sus expectativas. Esto sucedió durante un tiempo con Whatsapp, que dependiendo de la tienda donde se descargara, había que pagar o no por ella, aun cuando las prestaciones y características de la aplicación eran prácticamente idénticas en cada caso.

Independientemente de la plataforma donde se encuentre, el usuario paga por el valor, por algo que la app le aporte —que las demás no— y que justifique su precio, un precio que muchas veces, está condicionado por el mercado y por la competencia.

Un parámetro importante que suelen tener en cuenta los usuarios antes de pagar por una app es su valoración por parte de otros usuarios. Cuantas más y más altas valoraciones tenga, mucho mejor. Es más probable que un usuario pague por una app con muchas calificaciones y una media de 4.5 estrellas, que por una que no tenga ninguna valoración o que solo tenga valoraciones negativas.

Por último, es necesario saber que cada una de las tiendas —tanto Google Play, como App Store o Windows Phone Store— cobran una comisión de 30% sobre el precio de venta de la app. Esto quiere decir que el desarrollador se queda solo con el 70% de lo que el usuario pague por su descarga.

Freemium

El modelo *freemium* es una combinación de los dos anteriores. Su nombre viene de mezclar las palabras inglesas *free* y *premium*, y consiste en descargar la aplicación de forma gratuita, permitiendo al usuario un uso básico y limitado, con la posibilidad de recibir funciones más avanzadas, que se liberan previo pago.

En cierta forma, puede decirse que este modelo tiene lo mejor de ambos mundos: la app puede llegar a un número mayor de personas al ser gratuita, pero también permite ofrecer servicios avanzados a los usuarios que realmente la encuentran útil o quieren sacarle el máximo provecho.

La dificultad reside en determinar qué partes de la aplicación ofrecer de forma gratuita y por cuáles es conveniente hacer pagar al usuario. Por ejemplo, brindar demasiadas funciones gratuitas hará que poca gente quiera adquirir las pagas, pues la versión gratuita les resultará suficiente. Por otro lado, si la mayoría de funciones son de pago, los usuarios pueden encontrar la aplicación poco práctica y dejarán de usarla.

Un claro ejemplo de este tipo de distribución, son aquellos juegos que dejan avanzar ciertos niveles, pero para llegar hasta el final hay que pagar por la versión completa.

Monetización

Los modelos de monetización son diferentes caminos para obtener dinero a través de las aplicaciones. No deben verse de forma individual y separada, ya que suelen depender de si la app es gratis, paga o *freemium*, y también de la categoría de la aplicación.

Compras dentro de la app *(In-app purchase)*

Algunas apps permiten pagar pequeñas cantidades de dinero por la compra de ítems separados que mejoran las prestaciones básicas; por ejemplo, algunas aplicaciones de fotografía venden filtros de fotos avanzados. Es también el caso de las apps que ofrecen contenidos *premium* o suscripciones. Esta forma de monetización está más asociada a la distribución *freemium*.

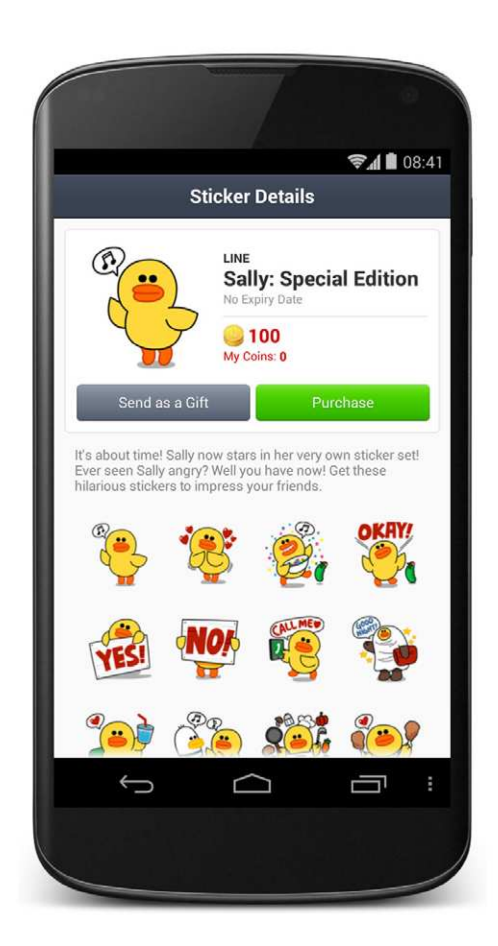


FIG 2.7. Line es una app que usa las compras in-app para vender pegatinas.

Pagar por la versión completa

En este caso, las aplicaciones se desarrollan en dos versiones: una gratuita con funciones básicas, que permite al usuario probarla con algunas limitaciones o publicidad y una de pago, que ofrece más posibilidades para quien esté convencido de la compra luego de haber probado la versión gratuita.





Fig 2.8.
Fruit Ninja es una app que tiene una versión gratis y otra de pago.

Este modelo está usándose cada vez menos por los inconvenientes que presenta tener dos aplicaciones separadas. Por un lado, los usuarios que quieren la versión *premium* tienen que descargar una nueva aplicación, con la dificultad que esto ocasiona, pues a nivel de desarrollo no siempre es fácil trasladar la configuración que ha definido el usuario de una versión a la otra.

Además, las valoraciones de los usuarios y la posición dentro de los rankings de las tiendas de aplicaciones son independientes para cada una de las versiones.

No obstante, estos inconvenientes pueden evitarse ofreciendo la versión *premium* como una compra dentro de la app.

Publicidad

La publicidad puede usarse en aplicaciones gratuitas como herramienta para obtener rédito económico. Suele presentarse en forma de pequeños avisos que pueden ser pulsados por el usuario para acceder a otras webs o descargar otras aplicaciones. En este modelo la ganancia depende de la cantidad de gente que entre a los anuncios.

Como principales inconvenientes se pueden nombrar la intrusión a la privacidad del usuario y que la visualización de avisos afecta la experiencia general.

Cada sistema operativo tiene su propio sistema de publicidad y en cada uno de ellos las condiciones serán diferentes. Google ofrece un programa de publicidad para las aplicaciones







FIG 2.9.
Cada sistema operativo tiene diferentes programas de publicidad para sus apps.

en Android llamado Google AdMob[12]; por su parte, para iOS se encuentra iAd[13] y para Windows Phone, Microsoft Advertising [14].

Al considerar la opción de incluir publicidad, es muy importante pensar en qué plataformas se quiere distribuir la aplicación. Los avisos pueden verse como algo normal en Android, ya que muchos de sus usuarios prefieren ver anuncios a pagar por una app, pero pueden ser un problema en aplicaciones para iOS donde los usuarios tienen un comportamiento diferente frente a la publicidad.

12 http://www.google.es/ads/admob/
13 https://developer.apple.com/iad/
14 http://advertising.microsoft.com/en-ca/windows-8-ads-in-apps

Para qué plataforma desarrollar

Antes de decidir si diseñar para una o varias plataformas, se debe tener en cuenta todo aquello que afectará el desarrollo, desde los recursos y la complejidad, hasta el tipo de usuario al que se quiere orientar.

Actualmente los sistemas operativos con mayor penetración en el mercado son Android y iOS, en ese orden[15]. Entre ellos se reparten la mayor parte del pastel, mientras el tercer puesto está un poco más disputado, con una participación activa de Windows Phone para ganarse ese lugar.

Mayor alcance o exclusividad

Diseñar para un sistema operativo popular como Android supone conseguir un alcance y una cantidad de usuarios potenciales mayor. Sin embargo, las diferentes resoluciones de pantalla y versiones del sistema operativo disponibles hacen más compleja la experiencia de diseñar para Android, pagando así —en cierta medida— el precio de estar disponible para más personas.

Por otro lado, diseñar para iOS significa concentrarse en un mercado menor y a la vez, más exclusivo. Apple no tiene la misma cantidad de usuarios que los móviles con Android —ni cerca— pero aun así, tiene la ventaja de ser más consistente en las resoluciones de pantalla y en las versiones del sistema operativo (es más fácil actualizarse). Por lo cual, estos factores no suponen un gran impedimento a la hora de diseñar.

Por su parte, Windows Phone corre con franca desventaja frente a estos dos gigantes, pero su crecimiento de la mano

15 Koestsier, John. Android up 13%, iOS down 7%, BlackBerry down 81% … and Windows Phone up a massive 52%. http://venturebeat.com/2013/04/01/android-up-13-ios-down-7-blackberry-down-81-and-windows-phone-up-a-massive-52/

de Nokia y HTC le ha dado algo de fuerza —recientemente Microsoft, el gigante detrás de Windows Phone, ha adquirido a Nokia—. Si bien, por su número de usuarios, ahora está fuera de la pelea, representa un mercado que a futuro puede llegar a crecer.

Entonces, si realmente se quiere diseñar una aplicación «para todo el mundo», está fuera de discusión que por lo menos debería estar disponible para Android y iOS.

Personalidad de los usuarios

Cada sistema operativo tiene usuarios con características —geográficas, demográficas, psicográficas y conductuales— que los diferencian. Aunque a primera vista parezca algo sin importancia, conocer el tipo de usuario da algunas pistas acerca de quien usará la aplicación y, sobre todo, qué espera de ella.

En general, puede decirse que los usuarios de iOS dan mayor valor a la experiencia de usuario, se interesan por los detalles y tienen un perfil socioeconómico más alto que los consumidores de otras plataformas. Por esta misma razón, están más habituados a pagar por las aplicaciones[16]. Alguien que usa iOS es amante de la consistencia, de ver cada cosa en su lugar y prefiere no encontrarse con demasiadas sorpresas. Esto se debe en gran parte a que Apple es un sistema más cerrado y restrictivo a la hora de aprobar las aplicaciones, estableciendo reglas de diseño que aseguran cierta calidad y regularidad en sus apps.

Android por su parte, es un sistema operativo de código abierto, libre para los aportes de usuarios y compañías que muchas veces le dan su toque personal. Esto define a un usuario más dispuesto a ideas nuevas, a aplicaciones que rompen paradigmas y que presentan alternativas para diferenciarse, lo

16 Jiménez Cano, Rosa. Google Play no puede con la App Store: http://tecnologia.elpais.com/tecnologia/2013/04/26/actualidad/1366971919_151325.html

cual a veces significa toparse con aplicaciones un poco caóticas, más allá de los últimos esfuerzos de Google por establecer con mayor claridad sus directrices generales de diseño. De la misma forma, al encontrarse en una mayor cantidad de terminales con variedad de precios diferentes, Android tiene un alcance más masivo.

Finalmente, Windows Phone es un sistema operativo que está atrayendo a los usuarios amantes de la simplicidad que transmite su interfaz plana y despojada de lujos. Se asocia más con la practicidad sobre la estética y por esto, parece centrarse en usuarios que prefieren una buena experiencia a través de una navegación simple.

Trabajar solo o en equipo

Afrontar un proyecto es un camino que puede tornarse bastante largo, dependiendo de la complejidad del desarrollo y del alcance que tenga la app. Producir una aplicación requiere de al menos dos personas: diseñador y desarrollador.

El diseñador será el encargado de definir, entre otras cosas, la estructura general de las pantallas y sus elementos de interacción, el diseño de la interfaz y la preparación de los archivos para enviarlos al desarrollador, quien a su vez, se encargará de que la aplicación deje de ser un conjunto de imágenes en pantalla, programando su funcionalidad.

Entre los dos pueden definir aspectos generales de funcionamiento, el alcance del proyecto y la experiencia de uso que se quiere conseguir con la aplicación. De hecho, trabajar en conjunto permite complementar los conocimientos de cada uno sobre el área del otro. Por ejemplo, un diseñador puede plantear una interfaz determinada, pero el desarrollador tiene que estar atento a este diseño para indicar la complejidad del desarrollo. Por otra parte, un desarrollador debe proponer la funcionalidad siguiendo los consejos de usabilidad del diseña-

dor. De esta manera, trabajando en tándem, puede obtenerse una aplicación de calidad.

Los equipos pequeños permiten una forma de trabajo ágil: siempre es más fácil y rápido ponerse de acuerdo con una persona que con dos o tres, pero de alguna forma limitan la calidad del proyecto y la complejidad a la que se puede aspirar.

El anterior es el grupo mínimo fundamental, pero si hablamos de ideales o de un *Dream Team*, el equipo puede hacerse bastante más grande, incluyendo un líder de proyecto que se ocupe de la coordinación general, especialistas en arquitectura de información y en usabilidad, diseñadores visuales, desarrolladores expertos en una plataforma, ilustradores, e incluso, redactores y personal de QA —control de calidad— que aseguren alcanzar la calidad deseada de la aplicación.

El inconveniente de un equipo más grande es que implica un mayor esfuerzo de coordinación entre las partes y una gran cantidad de gestión que se multiplica por cada participante. No siempre más es mejor, por lo que, cuando se incluyen más personas en el grupo, es necesario definir con detalle el papel de cada una dentro del proyecto, para que sea realmente de ayuda y no todo lo contrario.

Productos de excelente calidad, como iA Writer[17] de Information Architects, fueron desarrollados por empresas con relativamente pocos empleados. Estas mismas personas a veces ni siquiera comparten oficina. Los proyectos con formatos digitales pueden ser llevados a cabo por equipos distribuidos en diferentes partes del mundo, con herramientas de gestión de proyectos, como Basecamp, y, por supuesto, una buena comunicación.

17 http://www.iawriter.com/

RECURSOS

Antes de empezar, a la hora de decidir para cuál plataforma se va a desarrollar, es importante saber con qué se necesitará contar para llegar al ansiado día de publicación de la app, sobre todo para irse preparando y calcular la inversión necesaria. Y esto se refiere tanto a personas y conocimientos, como a equipamiento.

Si un diseñador lleva adelante el proyecto y necesita buscar a un desarrollador que complemente la sociedad, a veces no sabe exactamente a quién buscar y qué perfil debe tener este compañero. Hay que tener en cuenta que las diferentes plataformas requieren conocimientos de programación diferentes.

Android

Las aplicaciones de Android se programan en Java haciendo uso de librerías propias de Android, por lo que, a nivel de programación, un desarrollador con conocimientos sólidos de Java estándar no debería tener demasiados problemas para empezar a ser parte de la vida del mundo androide.

Para programar aplicaciones para este sistema operativo es indistinto tener un Mac o un PC —con Windows o con Linux—. Se puede descargar el software Android Studio y todo el material necesario para desarrollar una app desde la web de desarrolladores de Android[18].

En el momento del desarrollo, Android Studio permite usar los simuladores de diferentes dispositivos, o conseguir una prueba de funcionamiento más real conectando el terminal al ordenador

18 http://developer.android.com/

iOS

Un programador que quiera empezar a hacer magia desarrollando para iPhone e iPad debe tener una base de programación orientada a objetos, algo que le permitirá luego una transición más transparente a Objective-C, el lenguaje de programación que se usa en este caso[19].

A nivel de hardware y software para desarrollar aplicaciones para iOS, se necesita un ordenador Mac con el Kit de Desarrollo de Software —SDK— que en este caso es Xcode, el software oficial de Apple para desarrollo para iPhone e iPad, de descarga gratuita.

El código se puede probar directamente en el simulador —una representación del teléfono que permite ver como se comporta el código— dentro del ordenador, algo que sirve para la mayoría de los casos pero tiene ciertas limitaciones y no es completamente fidedigno, pues suele comportarse más rápido de lo que realmente lo hace en el teléfono.

Idealmente, para hacer una prueba más real de desarrollo, hay que probar el código en un iPhone conectado al Mac. Para esto es necesario pagar una licencia de desarrollador[20], algo que, de todas formas, hará falta más adelante para publicar la aplicación en la tienda. Este costo asciende a 99 dólares anuales.

Windows Phone

Un programador que haya estado trabajando en C# va tener un buen comienzo, ya que este es el lenguaje de programación que, junto con las librerías propias de Windows Phone, hacen una aplicación posible.

19 Kumar, Mugunth & Napier, Rob. http://iosptl.com
20 https://developer.apple.com/programs/ios/

Quien no sea muy fanático de Windows no va a tener otra opción que usar, por lo menos, Windows 7 en su ordenador para desarrollar. El consuelo es que puede instalarse tanto en un PC como en la máquina virtual de un Mac. Y hablando de software, también hace falta tener Microsoft Visual Studio. La versión gratuita es suficiente para desarrollar apps, pero está claro que pagando una versión completa se tiene acceso a muchas más comodidades.

El simulador de Windows Phone es relativamente bueno, porque la app se puede probar directamente desde el ordenador. Para tener una simulación más real, es posible conectar el teléfono al ordenador con Windows, si se tiene una licencia de desarrollo que cuesta 99 dólares anuales[21].

21 http://dev.windowsphone.com/en-us



3

Irene Pereyra: UX con magia en Fi

Debemos confesar que somos fans de Fantasy Interactive. Un lugar donde a muchos diseñadores les gustaría trabajar[1], un espacio para dejar volar la creatividad y rodearse de un gran equipo en alguna de sus oficinas alrededor del mundo.

La primera vez que supimos de ellos fue cuando su portafolio de trabajos cayó en nuestras manos, la mayoría de ellos, experiencias interactivas para clientes como Microsoft[2], Google[3] o USA Today[4]. Los casos de éxito de algunos de estos clientes están tan bien presentados y explicados en la web de Fi, con tanta atención a los detalles, que parecen proyectos en sí mismos. Este diseño cuidado se nota en cada una de las cosas que hacen, incluso en los íconos que usan en sus aplicaciones y webs[5].

A veces, muchos de nosotros buscamos una perfección que no llega porque no existe y, al final, nos deja sin sentirnos del todo satisfechos con el trabajo realizado. Pero entonces, ¿cuándo parar? ¿cuándo se termina un proyecto? Irene Pereyra, Directora Global de UX y Estrategia en Fi, se prestó para respondernos estas y otras cuestiones.

> *Hay un momento en cada proyecto cuando finalmente llega la claridad y dices ¡ajá! Todo lo que conduce a ello simplemente no se siente del todo bien y tu cerebro continúa procesando a través de varias iteraciones y formas de mejorar y simplificar. Hacer diseño de UX es un poco como desenrollar un ovillo de lana muy grande: no sabes cuánto tiempo te llevará, pero una vez que llegas al final, la satisfacción es inmensa. Ser capaz de dejar de iterar porque «está listo» es mi momento favorito de cada proyecto.*

1 http://blog.f-i.com/six-reasons-you-want-to-work-at-fi/
2 http://blog.f-i.com/explore-touch-with-internet-explorer/
3 http://www.f-i.com/google/ramayana/
4 http://blog.f-i.com/usatoday-com-redesigning-one-of-americas-most-popular-news-site/
5 http://www.f-i.com/fi/icons/

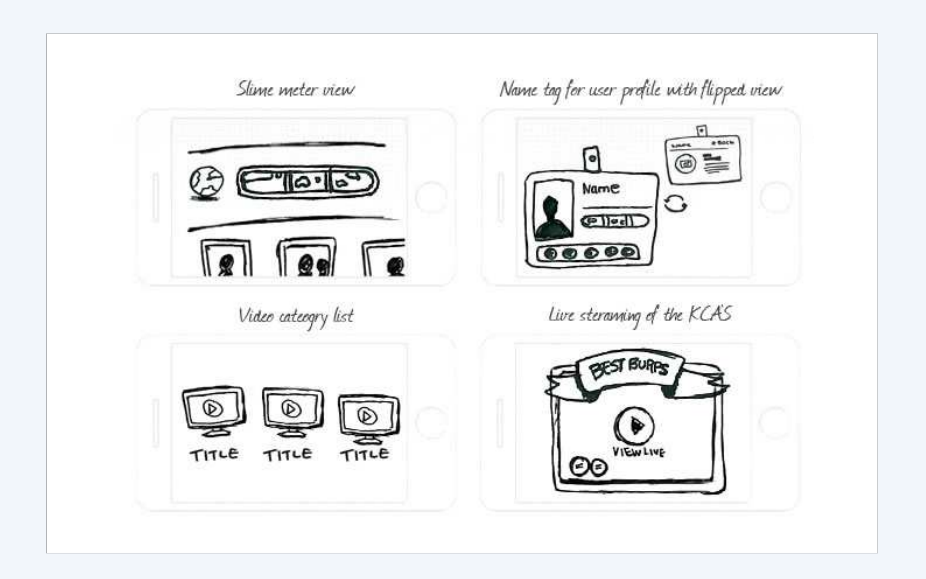


FIG 3.1. *Los wireframes, aun cuando sean en papel, son fundamentales antes de comenzar el diseño visual.*

Esa claridad de la que habla Irene, es algo a lo que se llega siguiendo las etapas de un proceso de diseño, que no comienza necesariamente con el diseño visual. Como comentamos en otros capítulos, la idea de una solución requiere ser bajada a la realidad —usando *wireframes* por ejemplo— antes de empezar a diseñar los detalles de una interfaz. Esto no es algo que muchos diseñadores suelen hacer. Entonces, ¿qué tan importante es?

> *Bosquejar y conceptualizar es tremendamente importante, ya sea que lo hagas en papel o en forma de wireframe en el ordenador, realmente no importa —¡idealmente deberías hacer ambos!—. No pasar directamente al diseño te permite enfocarte en cosas como la funcionalidad, la interacción de la estructura general y el viaje que quieres que el usuario haga cuando navega a través de tu experiencia interactiva. Omitir este paso es extremadamente peligroso porque te puede llevar a concentrarte solo en los detalles de diseño que, aunque son importantes, deberían*

considerarse en una etapa posterior del proceso, una vez que la estructura de interacción ha sido determinada.

La iteración es lo mejor durante esta etapa de «blanco y negro / servilleta de papel» porque aún no estás comprometido con ningún tratamiento visual y es mucho más fácil moverse rápidamente a través de diferentes conceptos e ideas en micro-iteraciones.

Irene ve el diseño visual como el resultado o la consecuencia de una planificación inicial de la experiencia de usuario:

Al final, conseguir una buena UX se trata de resolver lo que es apropiado para cada experiencia y factor de forma, y la capa de diseño que se aplica a esto, debería mejorar la experiencia interactiva general, sin importar si es skeuomorphic o plano.

Estas etapas iniciales son importantes para el proyecto si se logra una continuidad con el resto del proceso, que suponga la misma pasión por mantener la calidad. Y esto no es algo que se consigue fácilmente en proyectos en los que participan grandes equipos, con profesionales de distintos perfiles y además… el cliente.

¿Cómo se manejan como equipo y al mismo tiempo involucran al cliente durante el proceso?

Tenemos una aproximación interdisciplinaria a los proyectos, lo cual significa que todos los participantes están involucrados desde el comienzo con diferentes niveles de intensidad. Como tendemos a movernos bastante rápido por las diferentes fases, es útil tener las opiniones de los desarrolladores y diseñadores justo al

empezar el proyecto, pues tiende a aliviar potenciales problemas que puedan presentarse en el camino.

En lo referido a involucrar al cliente, además de los check-ins, reuniones y scrums normales, también publicamos nuestro progreso diario en formato PDF en Basecamp con vídeos explicativos. Nos encanta hacerlo para que el cliente pueda ver la presentación en su tiempo libre. Los clientes son personas ocupadas y no siempre tienen tiempo para leer todas las anotaciones; la posibilidad de ver el avance del trabajo cuando les resulta más conveniente, ha demostrado ser extremadamente efectivo para obtener retroalimentación.

Quizás no se piense en esto inicialmente, pero los clientes con los que trabajan también son, en cierta forma, elegidos por Fi, ya que al ser una empresa reconocida, muchas grandes compañías buscan sus servicios.

Han trabajado en proyectos para algunas de las marcas más grandes del mundo. ¿Alguna vez han dicho «no» a alguna de ellas? ¿Qué los motiva para comenzar un proyecto nuevo?

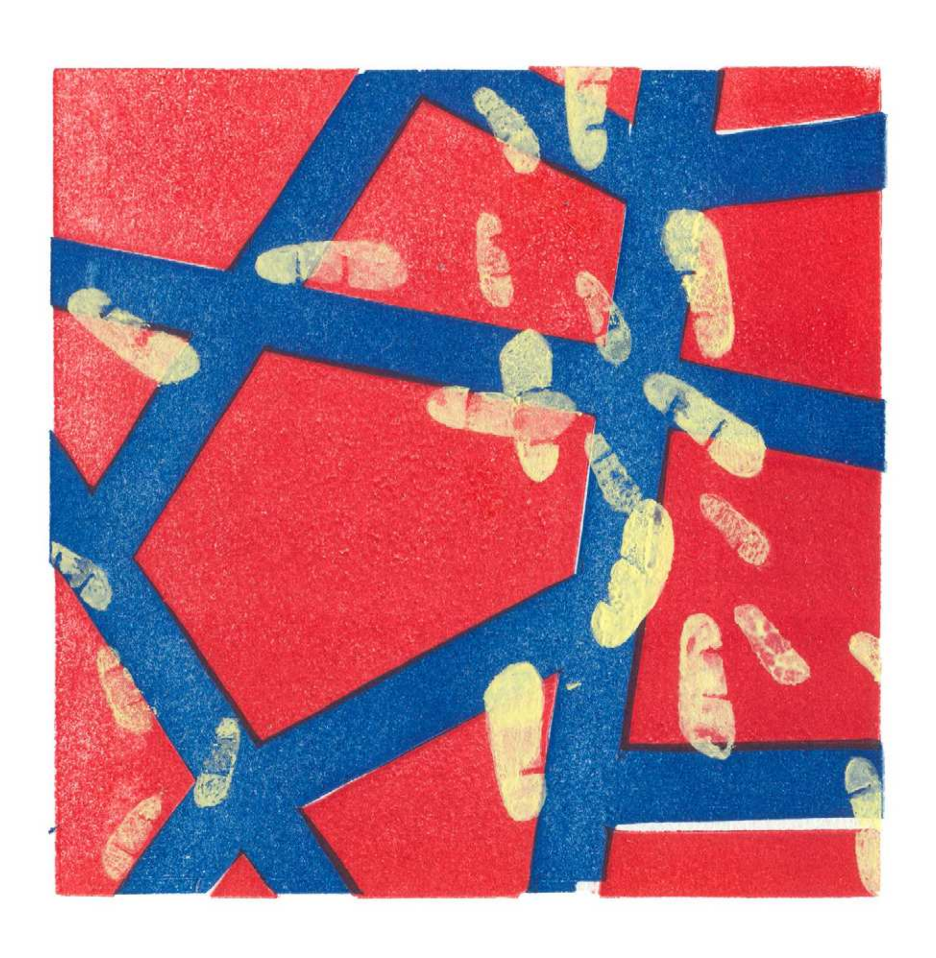
Sí, hemos dicho «no» a algunas de ellas, ¡incluso a algunas marcas grandes con las que quisiera trabajar la mayoría de compañías! La clave para un proyecto exitoso radica en tres puntos principales y, desde una perspectiva creativa, estas son las cosas que consideramos antes de tomar un proyecto nuevo:

1. *¿Está el cliente alineado con nuestra forma de pensar y está listo y preparado para ser innovador?*
2. *¿Se trata de algo que realmente no hayamos hecho antes? Si la respuesta es «sí», es genial, porque nos permite empezar el proyecto sin nociones preconcebidas o «equipaje».*

3. *Finalmente, ¿el presupuesto y el tiempo permiten experimentar y fallar? Nunca, en toda mi carrera, di en el blanco de inmediato y es muy importante saber que tienes el tiempo necesario para ir en la dirección equivocada por un ratito, antes de encontrar el rumbo de nuevo.*

   Lo que realmente me motiva para comenzar un proyecto nuevo es «lo gran desconocido». Me encanta sumergirme en cosas que me hagan sentir nerviosa e insegura y me gusta que a veces las fechas límite te fuerzan a ser creativo en formas que nunca habías imaginado.

Si quieres saber más sobre los proyectos de Fi, visita su sitio web y deslúmbrate: **www.f-i.com**



4

Explorando ideas

La cantidad de aplicaciones disponible en el mercado supone una gran competencia, pero también es un desafío y una oportunidad para sobresalir: ofrece algo que diferencie a tu aplicación y que tenga valor para el usuario. Una idea comprobada es el primer paso para conseguirlo.

Un mar de aplicaciones

En la vida cotidiana, es fácil encontrarse con situaciones que requieren una resolución inmediata, y al no hallarla, escuchamos frases como «debería existir una aplicación para esto». Esta forma de pensar, ha hecho que las diferentes tiendas estén saturadas de aplicaciones sin valor que no representan una utilidad real para el usuario. Se podría decir que hay apps para hacer casi cualquier cosa.

Este escenario es quizás más evidente en la tienda de Google, algo menos restrictiva a la hora de aprobar aplicaciones que se parecen entre sí: encontramos muchísimas apps de calculadoras o linternas que no aportan nada nuevo.

Sin duda, no se trata de un panorama alentador para el diseñador que va a lanzar su producto y representa un desafío para incorporar a la oferta una app que realmente se destaque y diferencie de las demás, ya sea por su utilidad real o por su experiencia de uso.







Fig 4.1. *Con diferentes propuestas, hay muchas aplicaciones de linternas.*

Puede que una app no sea la respuesta

Hay que considerar que una aplicación no necesariamente es la respuesta a cada circunstancia. Involucrar permanentemente la tecnología a veces fuerza la forma de resolver una situación, añadiendo agentes innecesarios en medio del proceso.

Por ejemplo, resulta inútil una aplicación para abrir la puerta del coche, cuando el hecho de sacar el teléfono del bolsillo, buscar la aplicación y la función adecuada para esta tarea, genera un camino más largo e incómodo para el usuario que simplemente usar sus llaves. Hacer algo usando una aplicación debería ser más fácil que hacerlo sin ella. El uso de las apps como respuesta a una situación se justifica cuando simplifica un proceso y mejora la experiencia para el usuario.

Valor para diferenciarse

Para diferenciarse de las demás, una aplicación tiene que aportar algo significativo y de valor para el usuario. El camino para empezar a conseguirlo se basa en tres pilares, que probablemente otras aplicaciones no han tenido en cuenta, y eso ya puede significar una diferencia.

1. Tener un objetivo

El objetivo es el motor de una app, lo que conduce todo. Prácticamente cualquier cosa que se incluya en una aplicación tiene que, en cierta medida, responder a él. Está directamente ligado a las necesidades del usuario y a la forma como la aplicación ayuda a resolverlas.

Las necesidades pueden estar relacionadas con el entretenimiento, la información, la interacción social o bien resolver un problema específico, como buscar el mejor camino para llegar a



Fig 4.2. El contexto de uso tiene en cuenta dónde están los usuarios al interactuar con la app.

casa cuando se está perdido. En la medida en que la aplicación responda mejor a ellas, será más valiosa para el usuario.

2. Pensar en el usuario

Las personas como centro de atención son la columna vertebral del llamado «diseño centrado en el usuario», que tiene en cuenta sus emociones, motivaciones y necesidades, a la hora de proponer soluciones.

La información que se obtiene al conocer al usuario permite tomar decisiones de diseño que ayuden a crear aplicaciones intuitivas y, por consiguiente, más fáciles de usar.

3. Determinar el contexto

El contexto de uso es el lugar donde se usará la aplicación: ubica al usuario en un espacio físico determinado que afecta y condiciona la forma que tiene de interactuar con la pantalla. El contexto también tiene en cuenta factores como el ambiente

general del lugar, las personas allí presentes y las acciones puntuales que en él se realizan.

Por ejemplo, no es lo mismo diseñar una app para una persona que estará corriendo o haciendo ejercicio en el gimnasio, que para alguien que estará sentado esperando el próximo bus o en casa enviando mensajes a sus amigos.

El nacimiento de la idea

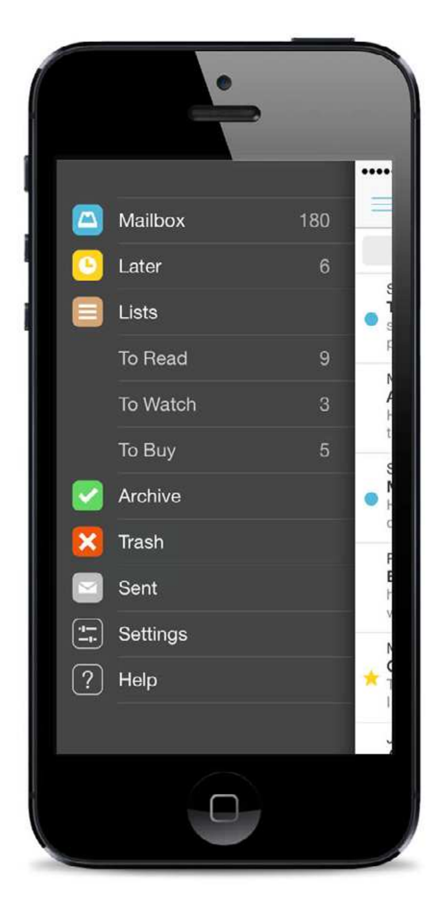
Las ideas cambian el mundo. Para un diseñador, una idea es una propuesta hacia los usuarios con base en su interpretación de cómo podrían resolver sus necesidades. Las ideas no tienen que ser necesariamente originales o únicas, pero sí tienen que ser mejores que las demás.

Posiblemente, al principio ni siquiera se tenga una idea. ¿Qué hacer entonces? Investigar. Y aunque esta palabra asuste un poco, puede ser tan simple como mirar alrededor: observar el entorno, las personas que nos rodean y las dificultades que encuentran en su vida cotidiana o las que encontramos nosotros mismos.

Qué hacer si ya existe algo parecido

Con una idea en la mano, o mejor dicho en la cabeza, el siguiente paso lógico es comprobar si esa idea ya existe, si fue desarrollada por otros anteriormente y de qué manera. Es lo que generalmente se conoce como *Benchmarking*.

Al investigar un poco por Internet o en las tiendas de apps, suele hallarse alguna aplicación parecida a primera vista, con la consecuente desilusión que esto ocasiona. Encontrarse en una situación como esta no quiere decir que deba abandonarse todo, dejar el proyecto de lado y volver a comenzar el proceso de pensar en algo.



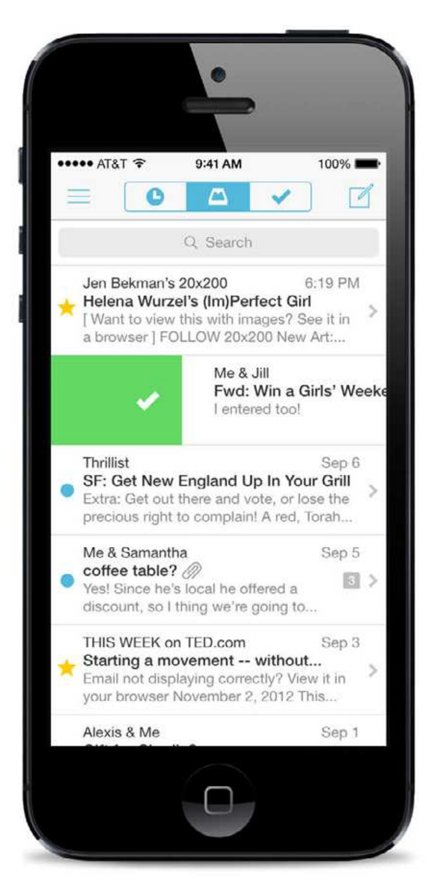


FIG 4.3. Mailbox es un cliente de correo que generó mucha expectativa por su forma de gestionar el correo como tareas para hacer.

Simplemente, se pueden analizar y valorar las alternativas existentes para entender cómo pueden ser mejoradas, complementadas u ofrecer algo que agregue valor a sus propuestas. En este caso, el camino para hacer la diferencia puede ser especializarse en alguna funcionalidad, trabajar en la simplicidad o en otros aspectos que, en definitiva, mejoren la experiencia de uso y la utilidad real de la app.

Más aún, toda esta investigación necesaria para mejorar la idea inicial puede ser un diferencial. Puede ayudar a darse

cuenta de algunas cosas que la eventual competencia no tuvo en consideración, convirtiéndose en algo que esta haya pasado por alto.

Esto fue lo que sucedió con Mailbox App[1], un cliente de correo. En el momento de su lanzamiento ya había otras alternativas en el mercado, como Gmail; sin embargo, aprovecharon la oportunidad de ofrecer un producto que cambiara el concepto de correo que tenían sus competidores y así consiguieron ser reconocidos.

Cómo saber si la idea funcionará

El siguiente paso después de tener una idea es realizar algunas comprobaciones para saber si realmente puede funcionar o si debe corregirse.

La forma de hacer esto es sondear con algunos usuarios que puedan dar su visión del concepto planteado. Es un proceso que no necesariamente tiene que ser costoso o largo, y que puede llevarse a cabo simplemente observando a las personas que nos rodean y tomando nota de sus costumbres y dificultades, evaluando cómo la idea propuesta podría ayudarlos. Otra forma más directa es comentar la idea con amigos o personas del entorno, para capturar rápidamente sus primeras impresiones.

Lo anterior no quiere decir que la investigación despeje todas las dudas o genere la totalidad de respuestas buscadas, pero sí sirve para tener una validación preliminar de la idea. Siempre es mejor basarse en información real que en meras suposiciones sin comprobar.

1 http://www.mailboxapp.com/



5

Definiendo la propuesta

Es hora de empezar a diseñar, al menos wireframes. Para ello, primero hace falta entender y estudiar un poco más a los usuarios, saber qué necesitan y cómo el diseño puede responder a sus necesidades.

Investigación del usuario

Conocer a los usuarios permite diseñar una aplicación que tenga en cuenta sus motivaciones, necesidades y problemas, como eje a partir del cual construir una propuesta. Este conocimiento no se basa en suposiciones y teorías, sino en estudios que ayuden a determinar el perfil de los usuarios de la aplicación. «Personas» y «Viaje del usuario», entre otras, son algunas de las metodologías utilizadas para conseguirlo.



Fig 5.1. Las Personas permiten extraer patrones comunes a muchos usuarios.

Las Personas

El concepto de «Personas» fue acuñado por Cooper[1], una compañía de diseño y estrategia ubicada en San Francisco, y es una herramienta de gran utilidad que se usa constantemente en el diseño de interacción actual. Su función es definir modelos o arquetipos de usuarios para los cuales diseñar, teniendo en cuenta sus necesidades y objetivos.

Para crear las Personas hace falta una investigación real que nos aleje de meras conjeturas, analizando muchos usuarios posibles y determinando cuáles son los patrones de comportamiento y pensamiento que tienen en común entre ellos, evitando caer en sus características individuales, centrándose solo en aquello que comparten.

El resultado final de esta investigación es una representación visual donde se modela al usuario a partir de los datos obtenidos: la Persona tendrá una cara, un nombre, una historia, ambiciones y objetivos.

Pueden determinarse diferentes tipos de Personas para una aplicación, pero para que este ejercicio tenga una utilidad real no deberían ser más de tres. Idealmente, el proyecto debería enfocarse en una sola Persona primaria.

El Viaje del usuario

Las Personas vistas individualmente pueden servir para conocer un modelo de usuario, pero también hace falta saber cómo se comporta y siente ese modelo, cuando tiene un objetivo que cumplir en un contexto de uso determinado[2].

1 http://www.cooper.com/journal/personas
2 Hobbs, Jason. An introduction to user journeys. http://boxesandarrows.com/an-introduction-to-user-journeys/



FIG 5.2. El Viaje del usuario se representa gráficamente para diferenciar sus etapas y la relación entre ellas.

Aquí es donde interviene el *user journey* o «Viaje del usuario», una forma de contar visualmente y de principio a fin, el proceso que lleva a cabo una Persona desde que tiene una necesidad hasta que la satisface usando la aplicación.

Nada mejor que un ejemplo para entender de qué se trata todo esto: alguien que está perdido y necesita orientación, puede usar el móvil para saber cómo llegar a casa. Para crear un Viaje de usuario a partir de esta situación, hay que considerar desde el momento en que está perdido, cuando aún no ha abierto la app, hasta que cumple su objetivo de orientarse, pasando por cada una de las acciones que lleva a cabo usando la aplicación.

Este proceso se puede visualizar gráficamente de una forma lineal, separando las etapas entre sí y detectando en ellas las emociones del usuario, pero también, las dificultades que encuentra en cada paso y las acciones concretas que realiza para seguir adelante, por ejemplo, buscar. De esta manera, se puede detectar en qué puntos debe enfocarse el diseño para resolver problemas de interacción y lograr una app más usable.

El Viaje del usuario sirve también para sentar las bases preliminares de la organización de la información y la definición de las funciones, sin pensar en una estructura rígida o jerárquica.

Definición funcional

Todas las acciones e interacciones que hacen falta para que un usuario consiga su objetivo, se traducen en funciones que debe tener la aplicación. Siguiendo el Viaje del usuario, se puede detectar cuáles son las necesidades que tiene en cada etapa y cuáles herramientas requiere para poder avanzar a la siguiente.

Continuando con el ejemplo de la aplicación para encontrar el camino a casa, serían entonces funciones fundamentales: determinar la ubicación actual, buscar la dirección de destino

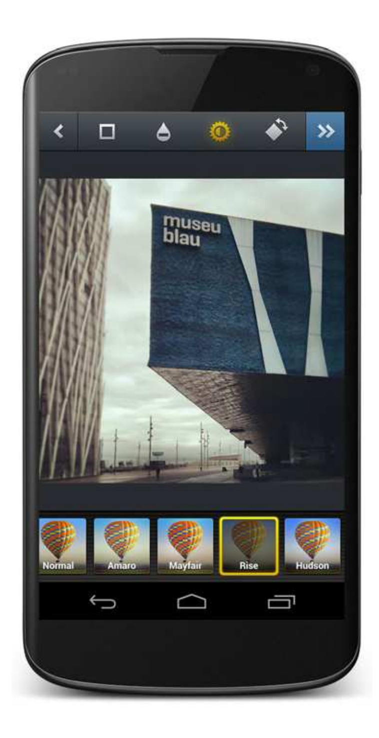


Fig 5.3. Instagram es actualmente una de las apps más populares para tomar fotos y videos.

y seleccionar entre las distintas opciones de transporte. Cada una de estas acciones es realmente importante porque ayuda al propósito de la aplicación.

Adicionalmente, pueden agregarse casi infinidad de funciones complementarias. Por ejemplo, una de ellas podría ser guardar la ubicación de destino para poder utilizarla después. Cada vez que se añada una nueva función que complemente el objetivo principal, hay que ser conscientes del valor que representa para el usuario y de su utilidad real, teniendo en cuenta la Persona y el contexto de uso. No hay que olvidar nunca lo que se espera de un móvil en cada caso particular, en nuestro ejemplo, la capacidad de resolver rápidamente tareas puntuales.

Cada función que se agrega representa, además, mayor tiempo de desarrollo y complejidad, de ahí la importancia de decidir con cuidado en cada caso si merece o no ser incluida en la aplicación, para no terminar con un producto saturado de funciones que nadie usa y que arruinan la experiencia de usuario. Siempre es preferible hacer pocas cosas, pero hacerlas bien.

Para poner un ejemplo del mundo real, Instagram[3], la popular aplicación de fotografías, llegó a ser lo que es ahora después de recortar una serie de funciones innecesarias, lo que le permitió concentrarse en el corazón del producto. Además, las funciones complementarias que añadió, como los filtros de fotos, representaron un valor real para el usuario y fue lo que le permitió diferenciarse de las demás.

Proponer una visión

Puede pasar muchas veces, sobre todo cuando el producto ya está en marcha, que los usuarios pidan funciones que consideran deberían incorporarse. Esto puede ser un arma de doble

3 Systrom, Kevin. Instagram: What is the genesis of Instagram? http://www.quora.com/Instagram/What-is-the-genesis-of-Instagram

filo, porque algo que sirva para un usuario particular no necesariamente sea útil para los demás[4]. En este caso, establecer un balance entre las opiniones de los usuarios y la propuesta de la aplicación será fundamental.

Arquitectura de información

La arquitectura de información es una forma de organizar el contenido y funciones de toda la aplicación, de forma que puedan ser encontrados rápidamente por el usuario. En sentido global, la arquitectura de información considera la relación entre los contenidos de diferentes pantallas y a nivel particular, la organización de contenidos dentro de la misma pantalla.

Luego de haber definido el Viaje de usuario y las funciones de la app, un diagrama de arquitectura utiliza esta información para determinar cuáles son las pantallas necesarias en cada etapa de ese viaje y qué funciones debería tener cada una de ellas. Además, aspectos de negocio y requisitos tecnológicos también pueden tenerse en cuenta para esta definición.

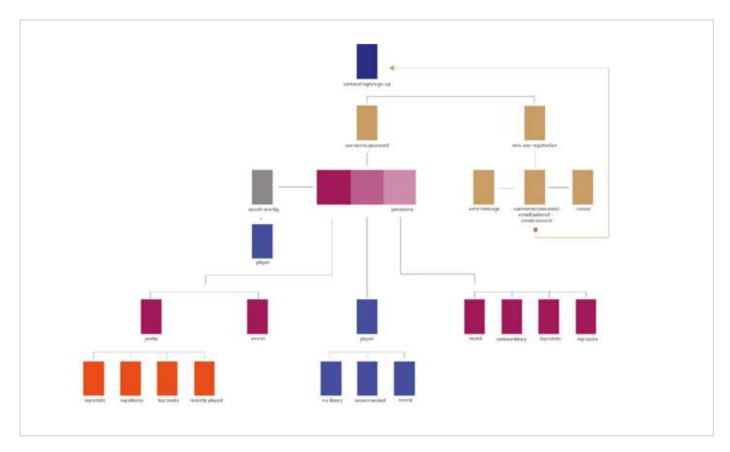


Fig 5.4. Un diagrama de arquitectura de información permite visualizar rápidamente los vínculos entre contenidos.

4 http://gettingreal.37signals.com/ch04_Make_Opinionated_Software.php

Una de las formas de visualizar la arquitectura consiste en representar cada pantalla con un rectángulo —en aplicaciones generalmente más vertical que horizontal— donde las conexiones entre los rectángulos indican la forma de navegar de una pantalla a otra y a través de qué acción.

Este diagrama sirve para estudiar la complejidad de la aplicación de un vistazo, analizar los diferentes niveles de profundidad, visualizar y entender la relación entre contenidos de una manera más organizada. Además, lo que se defina en esta etapa tendrá una incidencia directa en el tipo de navegación que se elija posteriormente.

Wireframes

Un *wireframe* —cuadro de alambre, según su traducción— es una representación muy simplificada de una pantalla individual, que permite tener una idea inicial de la organización de los elementos que contendrá, identificando y separando aquellos informativos de los interactivos.

Haciendo una comparación, para una aplicación, el wireframe es como el plano arquitectónico para una casa. En este plano se pueden ver los espacios y elementos funcionales de una forma clara y simplificada.

Tal como los planos de una casa, los *wireframes* están dibujados de forma lineal y del mismo color, lo que permite abstraerse de las tentaciones estéticas y centrarse en la estructura o esqueleto de una pantalla. Elementos tales como texturas, sombras y volúmenes se dejan de lado en esta etapa.

¿Para qué sirve un *wireframe*?

No siempre se hacen *wireframes*. Es normal que un diseñador se salte este paso para pasar directamente al diseño visual



FIG 5.5. Los wireframes son fundamentales para entender cómo puede estar estructurada cada pantalla de una app.

de la interfaz. Sin embargo, es una mala práctica que puede traer consecuencias después. Usarlos es más que recomendable porque sirve, entre otras cosas:

1. Como herramienta personal de exploración. Permite al diseñador evaluar diferentes alternativas de navegación e interacción, de una forma ágil y rápida, sin invertir demasiado tiempo en un diseño acabado que después no funcione.
2. Como herramienta para comunicar ideas abstractas. En etapas tempranas del proyecto, es necesario transmitir la idea general de la aplicación a otras personas, centrándose en la funcionalidad, objetiva y racional, evitando distracciones por la subjetividad de elementos estéticos. De esta manera, se puede tener un primer acercamiento a la app para trabajar en equipo.
3. Como un mecanismo para realizar las primeras evaluaciones de interfaz. Es posible detectar problemas de interacción y usabilidad, al sondear con usuarios o personas ajenas al proyecto, antes de haber diseñado o desarrollado aún una app funcional.

Hacer *wireframes* en etapas tempranas supone un ahorro de costes y tiempo, permitiendo evaluar aspectos como navegación e interacción para sentar unas bases sólidas que después den paso al diseño visual.

Formas de diseñar *wireframes*

Papel

Es la forma más básica y accesible. Se trata simplemente de tomar una hoja de papel y dibujar las pantallas y componentes de interacción con un bolígrafo. Generalmente todos los *wireframes* empiezan en papel y se van desarrollando y haciendo cada vez más complejos con otras herramientas, antes de diseñar la interfaz en software de ordenador.



FIG 5.6. Se pueden hacer wireframes en papel de una manera muy ágil.

Stencils

Este sería el paso siguiente y algo más «profesional» que un *wireframe* dibujado a mano alzada, manteniendo la esencia del trabajo en papel. Los stencils —generalmente de metal— con las que se pueden dibujar los elementos de interacción direc-



FIG 5.7. Con plantillas metálicas se obtienenen formas más definidas que a mano alzada.

tamente en el papel, con una estructura más formal, limpia y uniforme.

Plantillas

Esta manera de crear un prototipo, consiste en utilizar archivos digitales de plantillas que contengan los elementos básicos de interacción y de interfaz —botones, listas y cabeceras—, para construir en el ordenador, utilizando programas de diseño como Photoshop, pantallas básicas a partir de estos componentes.

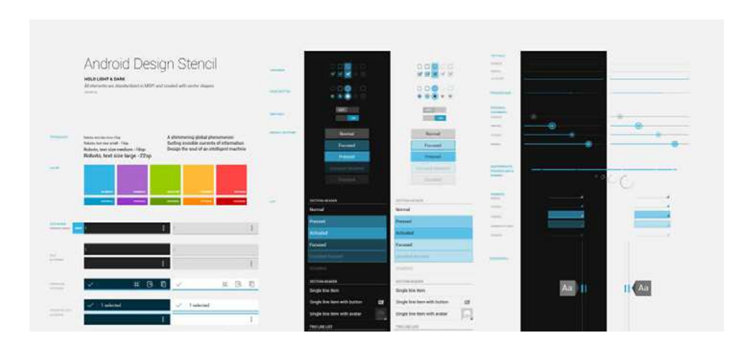


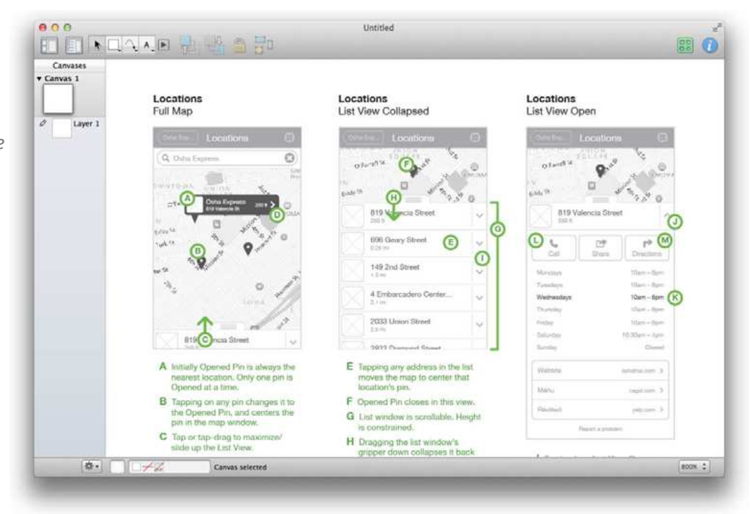
FIG 5.8. Existen plantillas –en este caso de Android– que se pueden aprovechar para diseñar wireframes en el ordenador.

Dependiendo del sistema operativo, se puede echar mano de diferentes recursos, siendo aquellos de iOS los más difundidos, seguidos de cerca por los de Android. Por otro lado, al día de hoy no existen demasiadas plantillas para Windows Phone.

Adicionalmente, también hay software de escritorio con librerías de plantillas para hacer *wireframes*. Entre los más conocidos se encuentran Balsamiq[5], Omnigraffle[6] y Axure[7], pero existen muchas otras opciones. Si se quiere diseñar *wireframes* usando el navegador y manteniendo las propuestas «en la nube», UXPin[8], permite trabajar de forma colaborativa con otros integrantes del equipo.

5 http://www.balsamiq.com/
6 http://www.omnigroup.com/products/omnigraffle/
7 http://www.axure.com/
8 http://uxpin.com/

Fig 5.9. Omnigraffle es un software de escritorio que permite usar plantillas importadas como librerías.



Eligiendo la forma adecuada

En realidad, no existe una mejor o peor forma de hacer *wireframes*. Las alternativas que ofrecemos son las más conocidas. Generalmente cuando se empieza a trabajar en diferentes proyectos, se elige la que sea más adecuada de acuerdo a la complejidad del trabajo a realizar. Es también una cuestión personal, de sintonía y de comodidad con la herramienta.

Hay diseñadores que se manejan muy bien dibujando en papel antes de pasar al ordenador y por el contrario, otras personas se sienten más a gusto trabajando directamente en el ordenador, llevando luego esas plantillas a los programas de diseño donde acabarán de darles forma.

Es recomendable probar diferentes opciones, hasta que se encuentre aquella que brinde comodidad, permita trabajar de una forma ágil y representar adecuadamente las ideas para las diferentes pantallas de la app.

Prototipos

Los prototipos son representaciones de la aplicación que sirven para probarla internamente o mediante test con usuarios, que permiten detectar errores de usabilidad en etapas tempranas de desarrollo. Generalmente, se trata de maquetas con una interacción suficiente para poder navegar entre las diferentes pantallas[9].

Pueden estar basados en *wireframes* o en diseños visuales, y su fidelidad puede ser mayor o menor, dependiendo de cuanto se corresponda su apariencia y comportamiento con la versión final esperada de la aplicación.

De qué pantallas hacer los prototipos

No es necesario hacer un prototipo que contenga todas las pantallas posibles de una aplicación. Los prototipos están destinados a pruebas, por lo tanto, deben desarrollarse solo aquellas pantallas necesarias para completar de principio a fin la tarea que se quiere probar. Por ejemplo, si se quiere sondear un registro de usuario, el prototipo debería contemplar las pantallas necesarias desde el ingreso de datos hasta el mensaje de éxito final.

Formatos de prototipos

Existen diferentes maneras de hacer un prototipo de aplicación: desde un dibujo en papel, pasando por el uso de software de diseño tradicional, hasta aquellos programas destinados exclusivamente al desarrollo de prototipos. La forma que se elija

9 Armengol, Daniel. ¿Qué es un prototipo? http://interactionphilia.com/que-es-un-prototipo

depende de la rapidez con que se quiera crearlo y la fidelidad respecto al resultado final esperado.

De acuerdo al formato de salida que se pueda obtener de un prototipo —para compartirlo posteriormente con otros miembros del equipo, clientes o accionistas— las formas de hacer prototipos pueden clasificarse en:

Documentos navegables

Los prototipos pueden presentarse en forma de documentos que se han usado toda la vida para presentaciones comerciales o distribución de información, añadiendo la interacción necesaria para navegar de una parte del documento a otro.

Por ejemplo, un PDF basado en archivos de diseño o fotografías de un boceto de la aplicación puede servir para representar el funcionamiento básico. Balsamiq es una herramienta que sirve para este fin.

Lo mismo sucede con archivos de presentaciones como los usados en documentos de Microsoft PowerPoint, Apple Keynote u OpenOffice. Keynotopia[10] permite crear, a partir de plantillas, documentos compatibles con estos software a los que se les añade una capa de interacción para simular el comportamiento de la app.

Para poder abrirse, este tipo de documentos tiene que ser primero descargado y las transiciones y animaciones entre páginas suelen ser algo básicas.

Versiones web

Muchos software de prototipos —de escritorio o en la nube— permiten crear representaciones de la app en forma de sitios web —y que por lo tanto necesitan conexión a Internet y un navegador— basados en HTML5 y CSS3 para conseguir interacciones y animaciones que se asemejan en gran medida a las de la aplicación.

10 http://keynotopia.com/

Algunos de estos programas permiten incluso crear un ícono de lanzamiento en la pantalla de inicio del teléfono, para conseguir una experiencia de simulación lo más cercana posible a la realidad.

Entre las herramientas que permiten crear prototipos de esta manera se puede nombrar a Codiqa[11], FluidUI[12], Framer[13] o Flinto[14], entre otros.

Otros formatos

Alternativamente, existe software específico para desarrollar prototipos. Uno con el cual se pueden conseguir resultados muy parecidos a los de una aplicación nativa es Briefs[15]. Utilizando este programa de escritorio puede crearse una simulación de aplicación, que necesita un visor especial —una aplicación gratuita— en el teléfono de destino para poder abrirse.

Por otro lado, si se quiere hacer un prototipo muy rápidamente y con el propio teléfono, se puede usar POP[16], una aplicación que permite usar la cámara del móvil para fotografiar *wireframes* en papel y dotarlos de interacción para navegar entre las diferentes pantallas. Sin dudas, una opción que puede sacarnos de apuros para realizar comprobaciones rápidas.

11 http://www.codiqa.com/
12 https://www.fluidui.com/
13 http://www.framerjs.com/
14 https://www.flinto.com/
15 http://giveabrief.com/
16 http://popapp.in/



6

Dustin Barker: La banca electrónica hecha Simple

Si pensamos en aplicaciones que realmente deben ser complicadas de diseñar y desarrollar, una de las primeras cosas que se nos viene a la cabeza es la banca electrónica. Las apps de este tipo tienen un montón de funciones posibles —pagos, transferencias, etc.— y además, no hay que olvidar el tema de la seguridad.

Simple, a secas, es un sistema de gestión de dinero basado completamente en una plataforma digital —no es un banco físico— que resuelve lo que podría ser muy complejo a través de una experiencia... sí, simple.

Por tratarse de un servicio que se utiliza enteramente en Internet, su aplicación tiene una importancia trascendental. Dustin Barker es el Director de Ingeniería Móvil en Simple, el encargado de desarrollar una app que, además de una experiencia de uso estelar, ofrezca al usuario una confianza tal, que no lo haga dudar ni por un segundo de la seguridad de la plataforma.

¿Cómo hacen para que sus usuarios se sientan seguros y confiados al usar la app de Simple?

> *Como los mecanismos que usamos para proteger la información de los usuarios no están a la vista, la sensación de seguridad tiene que venir de las interacciones. Aunque vamos «más allá» para proteger nuestras aplicaciones móviles, muchas veces son los detalles que no tienen nada que ver con la seguridad, los que hacen que los usuarios se sientan tranquilos.*
>
> *Brindamos confianza a través de la calidad. Una aplicación que responda y sea estable es fundamental, ya que, una alta calidad exterior le da una base a los usuarios para asumir lo mismo del interior de la app —y por lo tanto— mayor seguridad.*

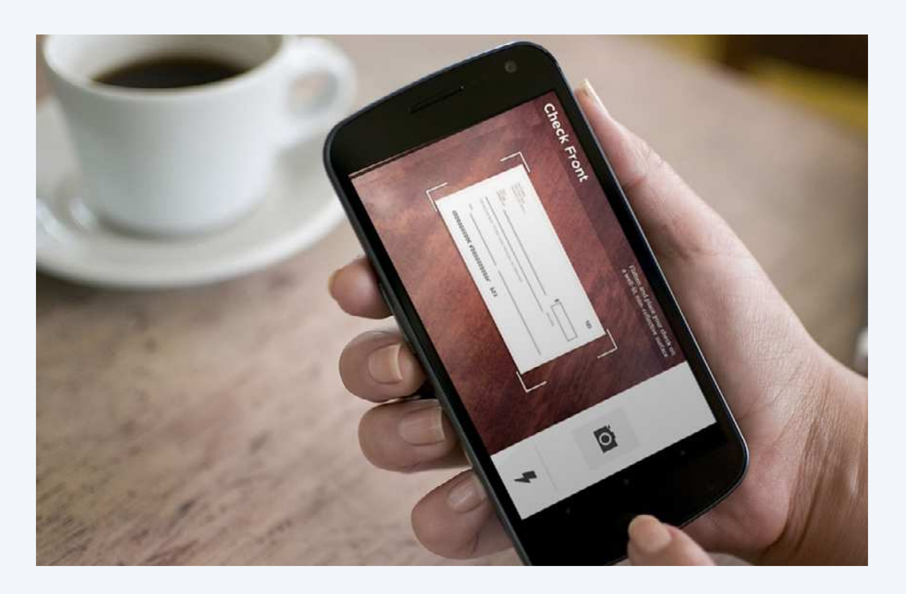


FIG 6.1. En Simple desarrollan en paralelo las aplicaciones para Android y iPhone.

> *Adicionalmente, pasamos mucho tiempo diseñando otros mecanismos de seguridad que los usuarios sí pueden ver. Funciones como la «autenticación multifactor» que usamos para enviar pagos, son esenciales no solo para proteger la interacción, sino también para construir la confianza de nuestros clientes mientras usan Simple.*

Dustin fue también ideólogo de la forma de trabajo *mobile first*, que privilegia el diseño de la app para móviles antes de hacer la web de escritorio; una metodología que, aunque últimamente está en auge, aún no muestra claras ventajas respecto al proceso tradicional.

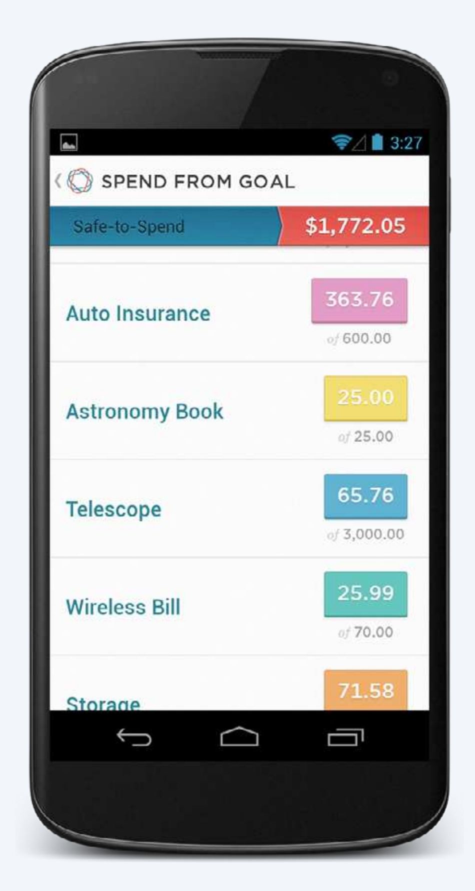
Si miras hacia atrás, ¿cuáles crees que han sido los beneficios más importantes de esta manera de trabajar?

> *Nuestro enfoque en móviles nos da ciertas limitaciones para trabajar y nos ayuda a tomar decisiones. Parámetros como el tamaño de la pantalla, la*

> *duración de la batería y la conexión a Internet nos obligan a reducir las funciones a lo esencial.*
>
> *Las aplicaciones para móviles se usan en contextos con distracciones —como esperando en una cola o durante una conversación—, entonces, usamos este conocimiento para tomar decisiones acerca de cómo hacer cada interacción tan eficiente e informativa como sea posible.*
>
> *Cuando pensamos acerca de una función, no nos limitamos solo a pensar en aquello que se encuentra entre la pantalla y el usuario, sino también en dónde se encontrará, qué podrá estar haciendo mientras interactúa con la app y cuáles son sus necesidades en diferentes contextos. Estas consideraciones nos ayudan a descubrir detalles acerca de una función que de otra forma podrían pasarse por alto.*

Y después de la aplicación para iPhone, ¿qué viene? Una versión para Android, claro. Dustin también nos cuenta lo que aprendió de la experiencia de desarrollar primero para iOS y lanzar, unos meses después, la versión para Android, dirigida a un tipo de usuario diferente:

> *Uno de nuestros objetivos principales al desarrollar Simple para iPhone era hacer todas las interacciones tan intuitivas como fuera posible. Para conseguirlo, nos apoyamos en el lenguaje que sabíamos que resultaría familiar para nuestros usuarios. Para lograr la misma meta en Android, claramente no podíamos usar el mismo lenguaje que en iOS. Tuvimos que empezar desde cero, así que re-imaginamos Simple usando solo arquitectura Android y el resultado fue una aplicación consistente con nuestra marca, que ofrece una experiencia única para estos usuarios.*





En las primeras versiones de Simple para iPhone, aprendimos a refinar y a simplificar cada función. También aprendimos cuáles aspectos de la app eran particularmente complicados o propensos al error. Si bien el diseño para Android fue completamente nuevo, pudimos empezar a construir a partir de Simple para iPhone.

FIG 6.2. *En este tipo de aplicaciones, ganarse la confianza del usuario es fundamental.*

Desde el lanzamiento de la app para Android, las decisiones de ingeniería que hemos tomado también han sido llevadas a la versión para iOS. Por ejemplo, sabíamos

que la lista de categorías en versiones anteriores de la app para iPhone estaba lejos de ser la ideal. Cuando estrenamos un diseño para esa lista en la primera versión para Android —una vez que vimos que tenía éxito— lo trasladamos a la aplicación para iOS y nuestros usuarios quedaron encantados con el resultado.

Ahora estamos en un punto donde podemos dejar que nuestras experiencias en ambas plataformas se enriquezcan mutuamente y podemos probar diferentes cosas para explorar qué funciona mejor en cada una.

Además de las aplicaciones para móvil, también está la web. En un servicio de gestión de dinero algunas operaciones pueden ser difíciles de resolver desde la pantalla de un teléfono y es mejor dejarlas para los ordenadores de escritorio, en la comodidad del hogar.

¿Cómo hacen para decidir cuáles funciones tienen sentido en el móvil y cuáles no?

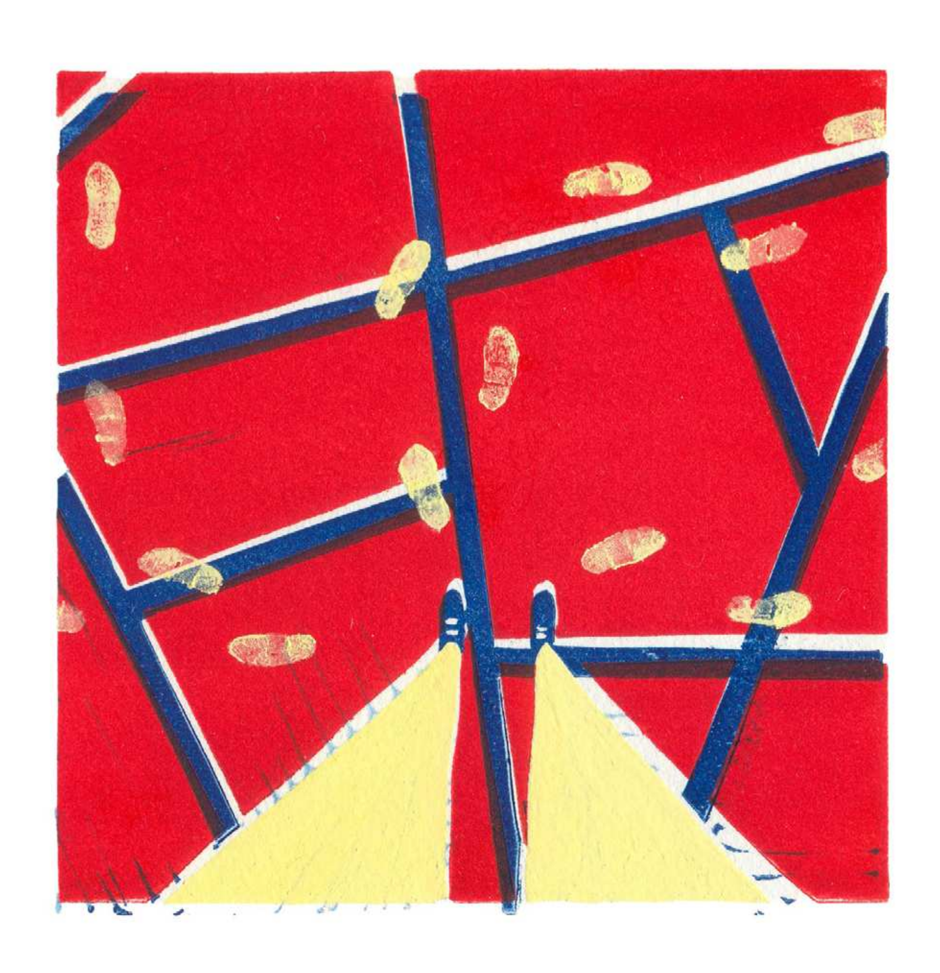
Las funciones de Simple pueden ser divididas en aquellas para analizar las finanzas —reportes o declaraciones— y las que sirven para controlarlas —enviar un pago, transferir dinero, depositar un cheque, etc.—; en estas últimas se enfoca nuestra app para móviles, pues son las funciones que nuestros clientes necesitan fuera de casa. Pueden leer un recibo, tomar el teléfono, abrir Simple y enviar un pago en segundos y en pocos pasos. Las funciones para analizar las finanzas podrían tener un lugar en la aplicación eventualmente, pero aún estamos buscando formas de optimizar estas interacciones.

Las aplicaciones de Simple son una gran inspiración para diseñadores y desarrolladores que sueñan con un producto con una

apariencia visual pulida, sin descuidar aspectos funcionales, ni de negocio. Contrario a lo que pueda parecer al principio, esto no es una quimera. Dustin revela su proceso de trabajo para conseguir un producto visualmente atractivo que funciona estupendamente:

> *La clave de Simple ha sido evolucionar continuamente el diseño a medida que se implementan las funciones. Nunca dejamos de diseñar la interfaz y no vacilamos si tenemos que revisar las decisiones, aun cuando el proceso de construir una función está muy avanzado.*
>
> *Cada función empieza con una etapa de diseño, durante la cual los ingenieros ofrecen comentarios —técnicos y estéticos— a los diseñadores. Una vez que tenemos una visión sólida, empezamos a desarrollar, sabiendo que la fase de diseño aún no está terminada. A medida que desarrollamos, descubrimos cómo «se siente» el diseño en una pantalla táctil y esto conduce a nuevas ideas acerca de cómo debería verse y comportarse la función. Para tener feedback constante, enviamos cada noche una versión a cada empleado.*
>
> *Siempre estamos buscando formas de mejorar cada función y nunca damos ninguna función por terminada. Esto solo es posible porque nuestros ingenieros y diseñadores trabajan en conjunto y colaboran constantemente.*

Aunque Simple está solo disponible en Estados Unidos, puedes ver su web para conocer más sobre ellos: **www.simple.com**



7

Interacción y patrones

Todos los sistemas operativos proponen diferentes formas de interactuar con los elementos en pantalla. Conocer la diferencia entre ellos y utilizar elementos familiares para el usuario, asegura que se sienta cómodo y seguro usando la aplicación.

Principios de experiencia de usuario

Cada sistema operativo tiene su propia identidad que es reflejada en la apariencia y comportamiento de cada uno de los elementos que componen su interfaz. En ellos imprime su personalidad, lo que hace que la experiencia sea diferente a las demás.

Sin embargo, todos comparten algunos puntos de vista fundamentales que se manifiestan en el diseño de sus interfaces. Los siguientes conceptos son considerados componentes clave del sistema operativo y de las aplicaciones que en él habitan.

Simplicidad

La simplicidad visual está directamente relacionada con la usabilidad. Ser simple implica en cierta medida ser mínimo, contar con pocos elementos, pero sobre todo, que aquellos presentes en la interfaz tengan una función bien definida que contribuya a cumplir el objetivo de la app y ayude al usuario.

Los móviles no son dispositivos para mostrar mucha información en pantalla. Por esta misma razón, la simplicidad consiste también en manejar la economía visual y tener un buen criterio para determinar qué incluir y qué no en el diseño. Una gran cantidad de elementos puede abrumar al usuario, por eso, lo que está en pantalla tiene que ser necesario en ese momento y en esa situación de uso.

Hacer el diseño simple es —vaya paradoja— bastante complicado, pero reporta grandes beneficios en la experiencia de uso de la aplicación.

Consistencia

Una app tiene diferentes pantallas que la componen y al mismo tiempo, está dentro de un sistema operativo que pro-

pone un determinado aspecto visual e interacción. El usuario de Android, iOS o Windows Phone ya está habituado a ellos y espera que las aplicaciones se comporten de la misma manera.

La consistencia, entonces, se trata de respetar estos conocimientos y costumbres del usuario, no solo en el interior de la aplicación, sino también en relación con el resto del SO. Esto favorece el uso intuitivo de la app, ya que el usuario puede prever su comportamiento sin demasiado esfuerzo.

La relación existente entre apariencia y comportamiento también tiene que ser consistente. El aspecto visual de un elemento interactivo determinado —como un botón con un ícono— puede llevar a esperar un comportamiento específico de acuerdo a la forma en que se ve.

Por ejemplo, si se usa un botón que representa la acción «eliminar» en el sistema operativo, el usuario esperará que también dentro de la app haga lo mismo. Cumplir con esa expectativa habla de consistencia.

Navegación intuitiva

Un aspecto que merece mucha atención en una aplicación es la forma de navegar entre contenidos, de manera que resulte fácil de comprender para el usuario, evitando la sensación de desorientación que puede ocasionar una navegación confusa.

La navegación intuitiva está también relacionada con la consistencia. Cada sistema operativo propone diferentes elementos para navegar por la app como botones, pestañas y paneles. Hacer uso de ellos hará que el usuario los reconozca a primera vista y, solo con estos componentes, ya sepa cómo ir de una sección a otra.

Por otro lado, para el usuario es importante saber y prever qué pasará después de pulsar un botón o cómo se mostrarán las pantallas. Intuir dónde se está dentro de los contenidos de la aplicación y conocer cómo volver hacia atrás son factores

muy importantes que alivian al usuario y le ahorran esfuerzos inútiles por intentar comprender cómo ir de un sitio a otro. Una navegación intuitiva permite, justamente, lograr un uso fluido y sin esfuerzo de la aplicación.

Interacción y formas de sostener el móvil

El diseño de aplicaciones para móviles tiene que tener en cuenta la forma en que los usuarios sujetan los teléfonos. Asimismo, con cuáles dedos interactúan y cómo los usan, tiene incidencia en el diseño de la interfaz y condiciona la ubicación de los elementos interactivos en la pantalla[1].

Si bien hay diferentes maneras en las que un usuario puede tener el móvil en sus manos, una de las más habituales es que lo sostenga con una sola mano, algo que puede ser liberador —ya que deja la otra mano disponible— pero al mismo tiempo,

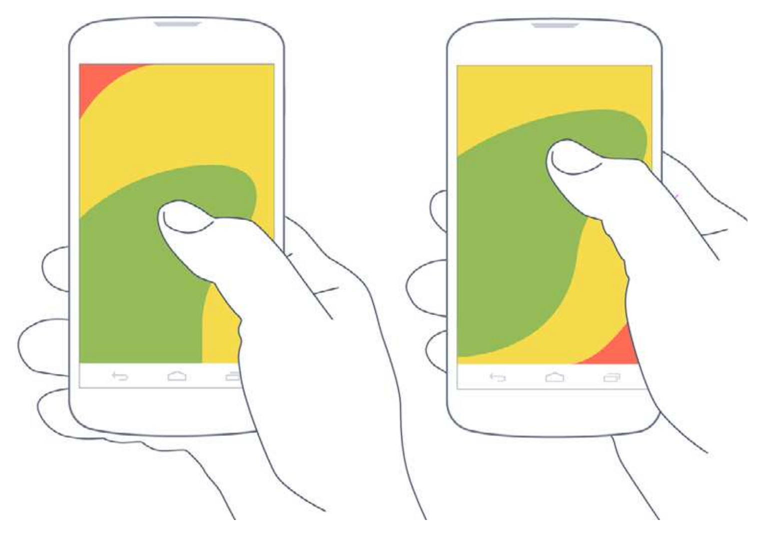


Fig 7.1. De acuerdo a la forma de sostener el móvil, el pulgar tiene mayor o menor dificultad en acceder a ciertas zonas.

1 Saine, Jamie. How Mobile Users Hold Devices. http://blog.utest.com/how-mobile-users-hold-devices/2013/03/

condicionante, porque otorga mucha responsabilidad al dedo pulgar para realizar las interacciones.

Entonces, las características anatómicas de la mano determinan ciertas zonas de la pantalla que pueden ser alcanzadas con mayor o menor comodidad por este dedo. La «Ley del pulgar»[2], mencionada por algunos autores, se refiere a la superficie de pantalla a la que este dedo tiene acceso sin mayores problemas y nos da pistas para organizar jerárquicamente los elementos en la interfaz.

Por ejemplo, los botones que se utilizan con más frecuencia deberían situarse en la parte inferior de la pantalla para ser alcanzados con facilidad, mientras que aquellos controles que no deberían tocarse por error —como «editar» o «eliminar»— se ubican fuera de esta zona, con un acceso más restringido.

De todas formas, tal como explica Josh Clark[3], la ubicación de los botones y otros controles en la zona inferior de la pantalla no se debe solamente a una cuestión de comodidad, pues ubicarlos en esta área también evita que sean tapados por la mano al desplazarse sobre el móvil. Este concepto ya se tenía en cuenta en otros objetos de uso cotidiano como iPods o calculadoras, donde el movimiento de la mano no interfería con la lectura de datos.

Sin embargo, cada sistema operativo hace un uso diferente de los espacios en pantalla y esto también condiciona el diseño de la aplicación. Por ejemplo, iOS sitúa las pestañas dentro de la zona de comodidad del pulgar, lo que permite un cambio fácil entre contenidos. Por su parte, Android utiliza las pestañas en el área superior —de más difícil acceso— para evitar conflictos con otros botones, como la flecha de «volver», ubicados en la parte inferior de la pantalla o del teléfono, algo similar a lo que sucede en Windows Phone.

2 Hinman, Rachel. Mapping the Screen for Touch. The Mobile Frontier. Rosenfeld Media.
3 Clark, Josh. Designing for touch. The Mobile Book. Smashing Magazine.

En teléfonos de pantallas grandes —más de cuatro pulgadas— o para realizar tareas de mayor precisión —como escribir— es bastante habitual sostener el dispositivo con las dos manos. En este caso, una mano se dedica solamente a sujetar el teléfono, mientras la otra —generalmente usando el dedo índice— realiza toques y gestos con mayor libertad y precisión.

Incidencia en la orientación del terminal

Tener en cuenta la orientación del dispositivo al utilizar la aplicación, significa aprovechar lo mejor de cada escenario. Generalmente, los teléfonos suelen sujetarse en forma vertical, mientras que en tabletas es habitual cambiar entre el formato vertical y horizontal con mayor frecuencia.

En los teléfonos, el modo horizontal se usa sobre todo en aquellas situaciones que requieren mejor aprovechamiento de la pantalla. Por ejemplo, sostener el teléfono en forma horizontal permite disponer de un teclado más grande y una mayor superficie para pulsar las teclas, permitiendo escribir más cómodamente.

Es recomendable diseñar para ambas orientaciones, ya que de esta forma no se fuerza al usuario a usar una única versión ofrecida. No obstante, en cada caso, habrá que evaluar si la

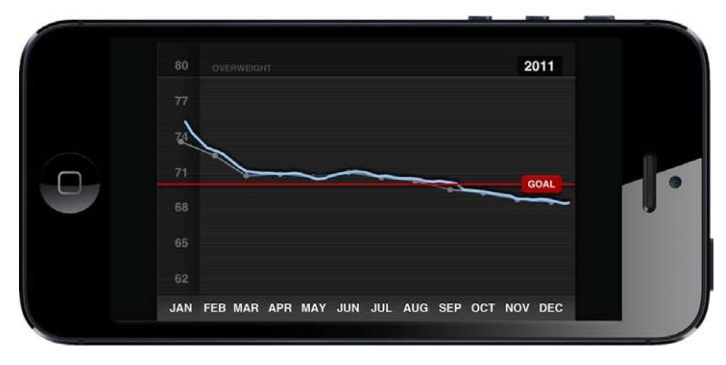


Fig 7.2. *Las diferentes orientaciones son una oportunidad de repensar la disposición de información que sea más útil en cada caso.*

aplicación realmente lo requiere, considerando que diseñar la versión horizontal no consiste simplemente en trasladar de manera directa cada elemento a la posición más parecida a su ubicación en vertical, sino en sacar el máximo provecho del espacio disponible en el modo apaisado, reubicando y acomodando los elementos gráficos e interactivos para mejorar la usabilidad.

Patrones de interacción

Según Martijn van Welie, «un patrón de interacción es un resumen práctico de una solución de diseño que se ha demostrado que funciona más de una vez. Utilízalos como guía, no como una ley»[4].

Entonces, los patrones de interacción son soluciones ya probadas para dar respuesta a problemas comunes de diseño, que ocurren una y otra vez. Apoyarse en ellos puede agilizar y simplificar el trabajo de diseño de una interfaz. Su utilización asegura, además, que los usuarios encontrarán elementos familiares en la interfaz que los harán sentirse más cómodos y seguros al usar la aplicación.

Navegación

Una navegación simple y consistente es un componente esencial en la experiencia de uso de la app y surge de contestar algunas preguntas básicas como: ¿De qué manera el usuario recorrerá la aplicación? ¿A través de menús o del contenido en sí mismo? ¿Y si viene de una notificación? ¿Cómo hará para volver atrás cuando haya avanzado?

4 http://www.androidpatterns.com/



Fig 7.3. *Las pestañas se ubican arriba en Android y Windows Phone y en iOS, abajo.*

Pestañas

Las pestañas —o *tabs*— se pueden utilizar para filtrar contenidos o cambiar entre pantallas que, de acuerdo a la arquitectura de información, tienen el mismo nivel de jerarquía, indicando siempre dónde se está y hacia dónde más se puede ir.

Las buenas prácticas indican que es necesario destacar siempre la pestaña seleccionada, mantener el orden y la ubicación inicial —que no cambien de pantalla en pantalla— y no usarlas para incluir otras acciones distintas a la navegación.

Android utiliza las pestañas en la zona superior de la pantalla —justo debajo de la barra de acción— que, a diferencia de iOS, generalmente se usan para el segundo nivel de navegación. Este SO tiene dos tipos de pestañas: fijas —se ven todas al mismo tiempo— o deslizantes —el usuario solo pude ver la pestaña actual y las dos adyacentes—. En cualquiera de los casos, Google sugiere aplicar una regla de oro: no emplear más de cinco o siete pestañas.

Por el contrario, en iOS se encuentran siempre en la parte inferior. En iPhone se puede mostrar un máximo de cinco pesta-

ñas y, en caso de necesitar más, la última de ellas se convertirá en una pestaña «más», donde se agrupan los apartados menos relevantes. Según las guías oficiales de Apple, las pestañas deberán estar siempre visibles desde cualquier sitio y se recomienda su uso para organizar los contenidos de las jerarquías más altas.

Windows Phone rompe con la metáfora visual de las pestañas con el denominado *Pivot Control*, aunque su función sea la misma. Se ubican siempre en la zona superior de la pantalla y por su gran tamaño juegan el doble papel de título y pestaña al mismo tiempo. La recomendación es usarlos de forma complementaria y subordinados al menú Panorama, que se ocupa de la navegación de las jerarquías de contenido más altas dentro de la aplicación.

Listas

Según como se mire, todo contenido estructurado dispuesto verticalmente puede conformar una lista. Esta forma de mostrar tantos ítems como sea necesario permite al usuario tocar alguno de ellos para obtener información complementaria.

Las listas pueden mostrar tanto textos como imágenes, pero es importante siempre jerarquizar su contenido. Por ejemplo,

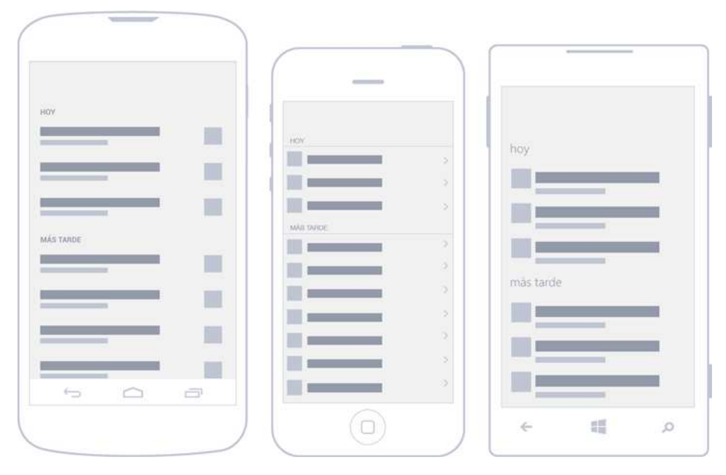


FIG 7.4. *Uso de listas en Android, iOS y Windows Phone.*

en una aplicación de correo electrónico es habitual darle mayor importancia al nombre del remitente que al asunto del correo, la fecha de recepción o las primeras palabras del mensaje. Ordenar los elementos dentro de una lista ayuda a visualizar el nombre del remitente —en este caso el dato más importante— de un primer vistazo.

Cuando contiene muchos elementos puede añadirse un sistema de índice que complemente la navegación a medida que se va desplazando, de forma vertical, por el contenido de la lista.

En Android el uso de listas está muy extendido. Las guías de estilo de Google recomiendan la agrupación de contenidos relacionados —tal como se usa en la pantalla de ajustes de ese sistema operativo— para una mejor comprensión, ya que como usuarios, somos capaces de memorizar a corto plazo solo unos pocos elementos.

En iOS las listas también ocupan todo el ancho de la pantalla. Si bien existen algunas que apilan un item sobre el otro de principio a finde la pantalla, existen también las listas agrupadas que permiten dividir los contenidos en bloques, separados por espacios verticales. Algo muy característico de las listas en iOS es que suelen incluir una pequeña flecha a la derecha de cada ítem.

En el caso de Windows Phone, el llamado *List View* muestra una lista de ítems que puede contener tanto imágenes como textos y añade la posibilidad de mostrar datos en formato de cuadrícula.

Galerías de imágenes

La disposición de imágenes está regida por la retícula propuesta por cada sistema operativo. En caso de que excedan el área disponible, se realiza un recorte —generalmente cuadrado— de las imágenes a mostrar.

Android es un caso particular, ya que al mostrar imágenes en la vista de cuadrícula, cuando se considere necesario, es posible utilizar un desplazamiento horizontal. Cuando esto

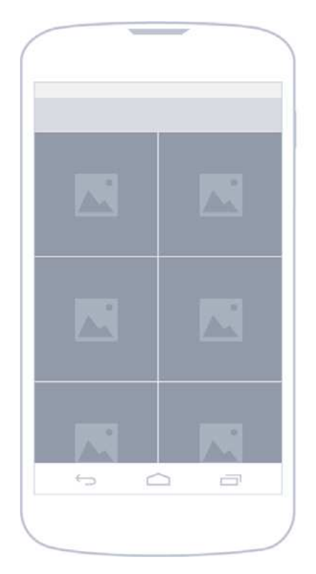
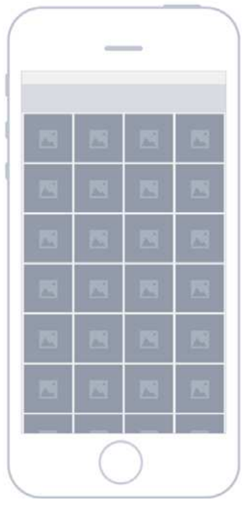
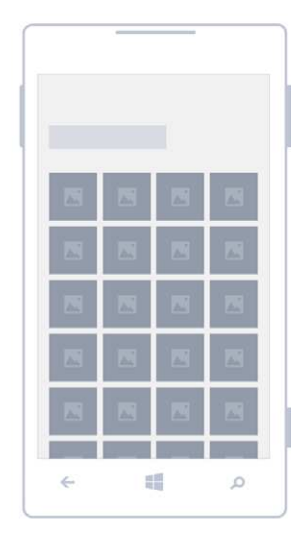




Fig 7.5. Las galerías de imágenes usan un formato reticular en Android, iOS y Windows Phone.

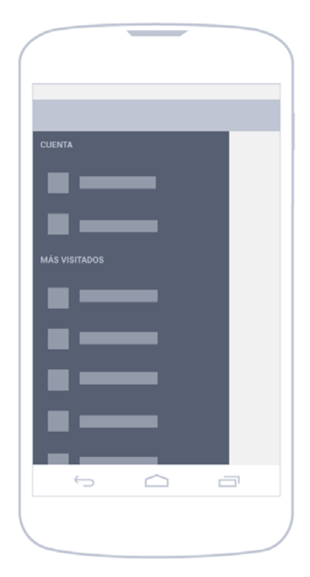
sucede es recomendable mostrar un pequeño trozo de las imágenes siguientes.

Menú tipo cajón

Este patrón, popularizado por Facebook, permite cambiar rápidamente entre pantallas de la aplicación. Pulsando un botón se despliega de forma lateral una lista con los contenidos, oculta hasta ese momento. Otra forma de llegar a esta lista es deslizando el dedo desde el lado izquierdo de la pantalla.

Las ventajas del uso de este patrón están claras: mejor aprovechamiento del espacio y, una vez desplegada la lista, ofrece una forma cómoda de navegar los contenidos. Pero no todas son buenas noticias, ya que este tipo de menú obliga a los usuarios a tocar el botón y desplegar el panel para poder saber cuáles son las opciones disponibles en la lista.

Hasta el momento, Android es el único que ha estandarizado su uso en las guías oficiales, recomendándolo para los niveles de navegación más altos de la app o cuando las opciones de



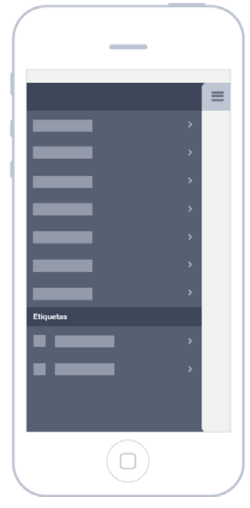




FIG 7.6. *El menú tipo cajón se ha hecho muy popular en Android, iOS y Windows Phone, aunque por ahora solo el primero lo incorpora en sus guías oficiales.*

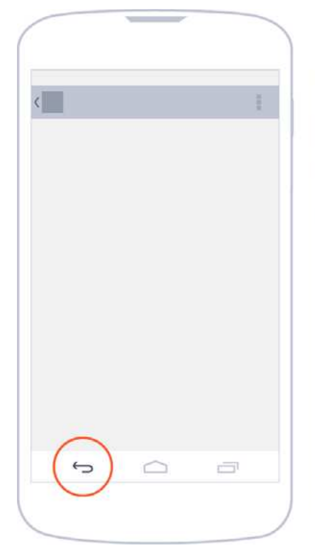
menú no tengan relación directa entre sí —en el caso contrario es mejor utilizar pestañas—.

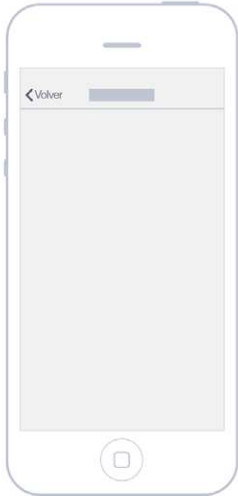
En iOS se pueden encontrar muchas aplicaciones que lo utilizan. Un ejemplo de buenas prácticas —aunque no reconocido por las guías de diseño— es Path. Esta red social ha implementado este tipo de navegación desde su versión 2.

Por otra parte, en Windows Phone su uso aún no está muy extendido. De hecho, la aplicación de Facebook es una de las pocas que utiliza el menú tipo cajón para navegar por los contenidos.

Volver

Los usuarios acostumbrados a usar la web encontrarán aquí una forma conocida de navegar. A medida que se va avanzando en profundidad de contenidos es necesario contar con una forma de retroceder o volver a niveles superiores. En el mundo móvil, con la navegación pantalla a pantalla, el uso del botón «volver» es muy frecuente.





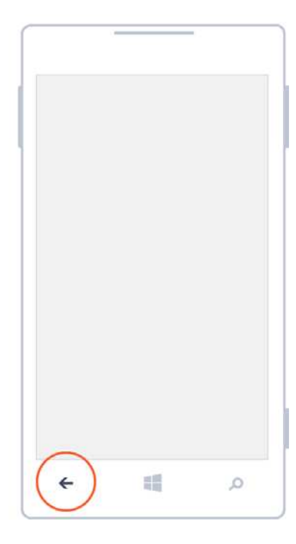


FIG 7.7. El botón de volver en Android se usa arriba —entonces llamado up— y abajo; en iOS, siempre arriba y en Windows Phone, es el botón físico del terminal.

En iOS, este botón está contenido en la barra de navegación y se ubica en la esquina superior izquierda con una etiqueta que tiene el título de la página anterior. Aquí la navegación entre páginas es jerárquica.

En el caso de Windows Phone, el botón físico del teléfono —fuera de la interfaz de la app— se encarga de gestionar esta forma de navegar entre los contenidos. De hecho, la interfaz de la aplicación no debería contemplar el uso de este botón, dejándolo en manos del sistema operativo.

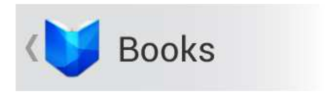
Android y la confusión… ¿arriba o atrás?

A partir de la versión 4, Android propone una nueva manera de navegar la estructura de información de las apps, basada en la relación jerárquica entre pantallas y para esto, ha introducido el botón «arriba», en inglés *up*. En la pantalla de inicio de la app, entonces, no debería aparecer porque no hay niveles superiores.

Por su parte, el botón «volver» es físico en algunos teléfonos e incorporado a la barra de navegación virtual —en la interfaz del sistema— en otros terminales. Está visible en todo momento

FIG 7.8. Android dispone actualmente dos formas de navegar hacia atrás.





y se utiliza para navegar hacia atrás —cronológicamente— por el histórico de pantallas por las que ha pasado el usuario. Si se pulsa este botón estando en el nivel superior de la app, el usuario terminará por salir de ella.

Estos dos botones trabajan de forma complementaria, pero en ciertas ocasiones pueden representar conflictos de navegación que hay que considerar en cada caso particular, para ofrecer al usuario la mejor experiencia posible.

Windows Phone y la navegación Panorama

Panorama es un patrón exclusivo de Windows Phone que ofrece la posibilidad de navegar los contenidos de forma horizontal, como si cada una de las pantallas estuviera colocada una al lado de la otra. El desplazamiento por estas pantallas da la sensación de continuidad entre los contenidos apoyada, en parte, por la utilización de imágenes de fondo.

El uso de Panorama está recomendado para el nivel superior de navegación, a modo de página de inicio. Desde allí se dan a conocer los principales apartados de la app, como pequeños avances del contenido que se encontrará luego con más detalle.



FIG 7.9. Panorama, una forma de navegar entre pantallas de primer nivel.

Este patrón se compone de una imagen de fondo, título —generalmente el nombre o logo de la app—, títulos de secciones y los avances de cada una de ellas.

Acciones

¿Qué acciones son necesarias en este momento? ¿Qué acciones esperaría encontrar el usuario luego de acceder a esta pantalla? ¿Cuál de todas las acciones es la más importante? Estas son algunas de las preguntas que hay que hacerse a la hora de definir las acciones que se encontrarán en cada pantalla de la aplicación.

La gran mayoría de acciones se pueden realizar en determinadas páginas y solo allí tienen sentido. Sin embargo, puede haber excepciones cuando sea necesario que una acción esté siempre visible. En este último caso, las acciones realmente importantes deben ser destacadas de una forma muy evidente. Es lo que sucede en Twitter con la acción para escribir un nuevo *tweet*, presente en la interfaz la mayor parte del tiempo.

Hay diferentes sitios donde ubicar las acciones de acuerdo a su jerarquía y funcionamiento, donde las más importantes están visibles y las menos, ocultas. Donde sea que se encuentren, la posición tiene que ser consistente a través de las distintas pantallas y con otras aplicaciones del sistema operativo.

Barra de acciones

En todos los sistemas, el compendio de acciones que se pueden realizar se representa por medio de íconos, por ello la correcta selección de estos recursos gráficos es fundamental.

En Android, los botones de acción se encuentran en la zona superior derecha de la interfaz; aunque puede haber excepciones donde se ubiquen en la parte inferior de la pantalla, separados de la navegación. Es fundamental ordenar las acciones en función de la frecuencia de uso. El ancho de la pantalla

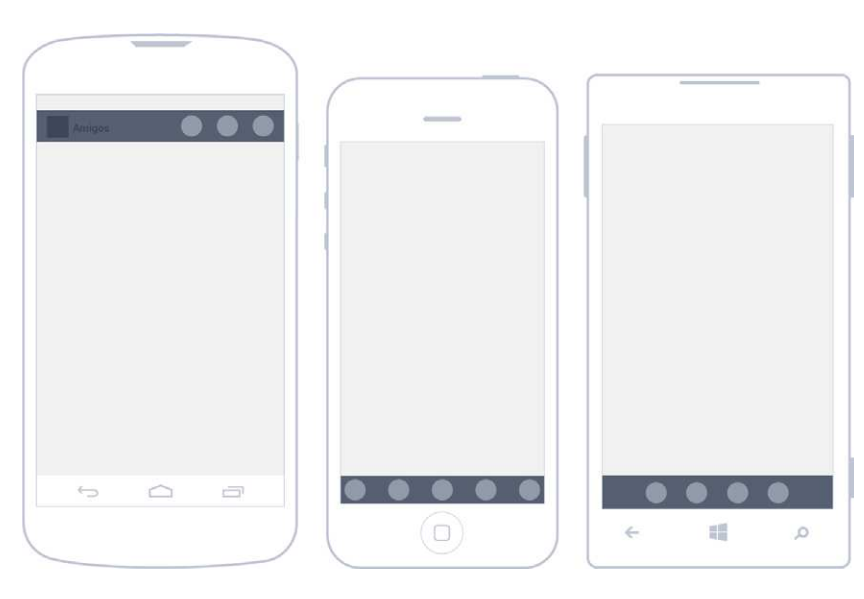


Fig 7.10. La barra de acciones se ubica arriba en Android y abajo en iOS y Windows Phone.

determinará cuantos ítems se pueden mostrar: desde dos en los móviles más pequeños, hasta cinco en tabletas.

En el caso de iOS, lo más común es que las acciones se ubiquen en la zona inferior, aunque en iPad se colocan en la parte superior derecha.

Igualmente, Windows Phone utiliza siempre la zona inferior de la pantalla para localizar las acciones que se pueden realizar y desde la documentación oficial insisten que solo deberían mostrarse las más comunes —cinco como máximo—. También puede pasar que no haya nada destacable, caso en el que es conveniente usar la barra minimizada.

Desborde de acciones

Las funciones extra y de uso poco frecuente se descubren por medio de la «revelación progresiva». Básicamente, están ocultas la mayor parte del tiempo, hasta que el usuario las reclame.

En Android, las opciones que no caben en la barra de acción pasan automáticamente a mostrarse como acciones desbordadas. El camino para llegar a ellas es a través de un botón con







Fig 7.11. Android, iOS y Windows Phone plantean diferentes alternativas para resolver el desborde de acciones.

un ícono de tres cuadrados verticales que las abre en formato de lista.

La propuesta de Apple consiste en agrupar acciones relacionadas, ocultas inicialmente, para luego mostrarlas en formato de lista de botones.

En Windows Phone, al igual que Android, las acciones extra se ubican por debajo de la barra de acción y se indican con puntos suspensivos que, al pulsarlos, despliegan las acciones ocultas en formato de lista.

Accesos rápidos

Hay ciertas acciones que deben estar muy a mano para que los usuarios puedan alcanzar sus objetivos rápidamente, por ejemplo, acceder a las acciones asociadas a ítems en una lista o retícula sin tener que navegar en profundidad para encontrarlas.

En el caso de una app de música, cuando se quiere añadir una canción a la cola de reproducción, sería un fastidio tener que entrar a la página de detalle para hacerlo. En situaciones

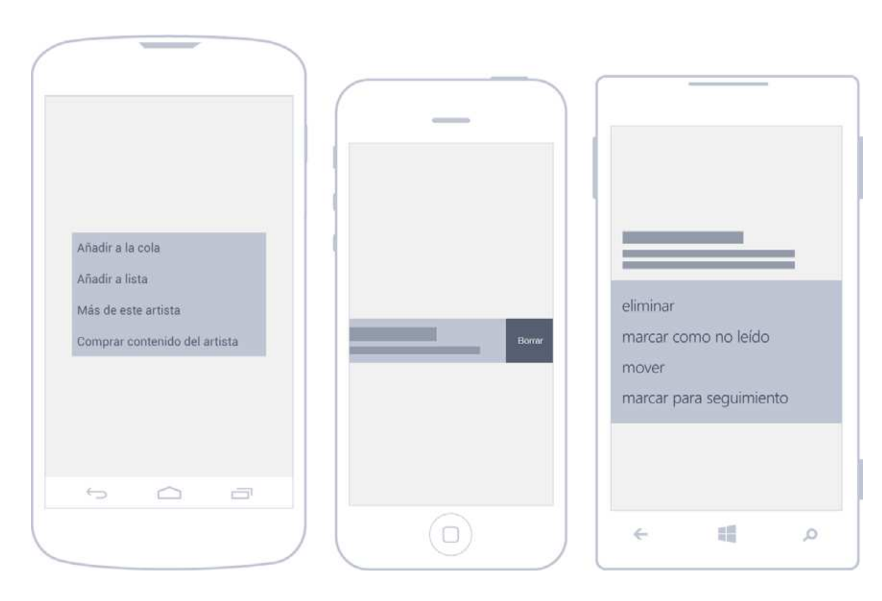


FIG 7.12. Formas de acceder rápidamente a acciones en Android, iOS y Windows Phone.

como esta, para simplificar estas acciones repetitivas es aconsejable utilizar los accesos rápidos.

En versiones anteriores a Android 4, se llegaba a las acciones rápidas pulsando de forma continua un ítem, lo que ocasionaba que se mostrara un menú con las acciones posibles. Actualmente, ese gesto —mantener pulsado— está recomendado solo para acceder al modo de edición de una lista; entre tanto, el acceso rápido es posible mediante un ícono triangular ubicado al pie de los elementos que tienen este tipo de acciones asociadas.

Un ejemplo bastante común de acceso rápido en el caso de iOS, es cuando se quiere eliminar un ítem de una lista y se realiza un deslizamiento horizontal sobre la fila deseada. Por otro lado, las acciones relacionadas se ubican in situ, sobre el mismo ítem.

En Windows Phone, en cambio, se despliegan en formato de menú contextual al efectuar el gesto de mantener pulsado sobre un ítem.

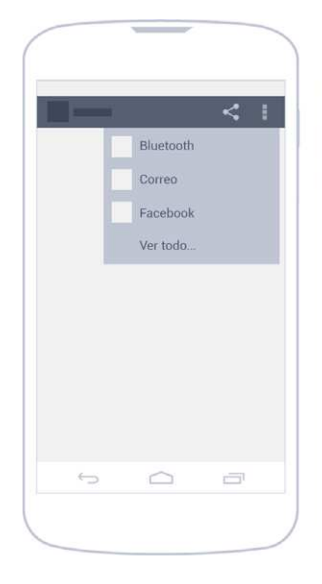






FIG 7.13. La acción de compartir se despliega arriba en Android, abajo en iOS y en pantalla completa en Windows Phone.

Compartir

Probablemente sea una de las acciones más empleadas en estos días: compartir contenidos con amigos, en Facebook, en Twitter, por mensaje de texto, como sea. Los sistemas operativos también han notado esta necesidad y han facilitado una implementación integrada al sistema muy fácil de aprovechar.

Buscar

Teniendo en cuenta que uno de los usos principales del móvil es el consumo de contenidos, la herramienta «buscar» es una manera esencial de llegar a ellos. En apps que muestran grandes cantidades de datos, la búsqueda puede ser incluso la función primaria.

La búsqueda se puede llevar a cabo mediante la introducción de texto —el método más habitual— o bien, por voz. Siempre que sea posible, es preferible ir mostrando los resultados a medida que el usuario escribe para mejorar la experiencia de uso. Idealmente, el tiempo de espera entre la introducción de los datos y el resultado no debería ser superior a uno o dos segundos.

FIG 7.14. *Buscar se ubica arriba en Android y iOS; en Windows Phone utiliza una pantalla aparte.*



Android hace que la opción para buscar sea accesible desde la barra de acciones. Si la búsqueda es una característica importante para la aplicación, como en el caso de Dropbox[5], debería ocupar la primera posición de esta barra. Al pulsar «buscar», la barra superior se modifica para convertirse en la de búsqueda.

En iPhone es habitual encontrar un campo de búsqueda por encima de las listas, tal como aparece en Contactos y otras apps. Junto con el campo de texto pueden aparecer filtros para refinar las búsquedas más complejas.

En Windows Phone, la búsqueda es tratada como una acción más, disponible en la barra de acciones y se lleva a cabo en una página separada, donde se introduce el texto y se listan los resultados.

Edición de listas

Es posible que el usuario necesite modificar varios elementos de una lista de forma simultánea. El flujo para realizarlo es bastante sencillo: se seleccionan los elementos sobre los que se quiere actuar y luego se aplica la acción correspondiente.

5 https://www.dropbox.com/mobile

Por ejemplo, si en una lista de correos recibidos se quiere añadir una etiqueta a tres de ellos, hay que marcarlos y aplicar la acción deseada. Hasta aquí nada del otro mundo.

Ahora viene lo más interesante: la manera de seleccionar elementos en una lista varía considerablemente de un SO a otro. ¡Que no cunda el pánico!, lo explicamos a continuación.

La selección múltiple se realiza en Android manteniendo pulsado un elemento. Una vez seleccionado el primer ítem, la barra de acción cambia indicando cuántos elementos están seleccionados y qué acciones se pueden realizar. Para salir de esta vista, Android propone hacerlo con el botón «check».

En iOS, la edición de listas suele activarse por medio de un botón «editar» colocado en la barra superior. De esta forma, la selección se hace visible junto con las acciones relacionadas. Para salir de esta vista sin aplicar los cambios, en el mismo sitio por donde se ha accedido, ahora hay un «Cancelar».

En Windows Phone hay dos maneras de acceder a la selección. Puede ser a través de la barra de acciones donde el usuario encuentra la opción «seleccionar» o usando un atajo que resulta muy práctico una vez aprendido: el truco está en tocar el primer ítem en su lateral izquierdo. De esta forma se hace visible la

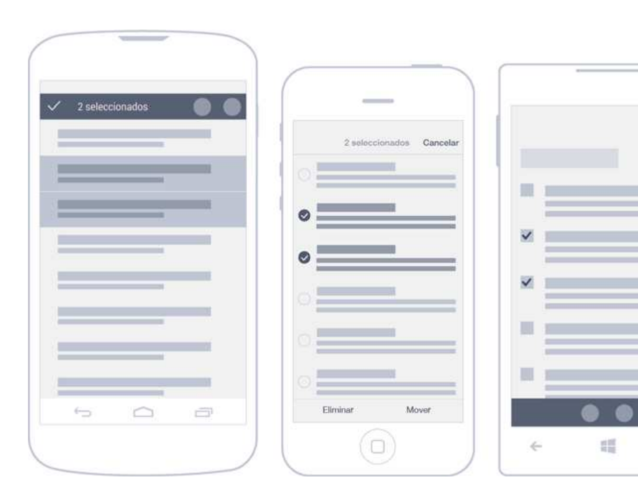


FIG 7.15. Tanto Android, como iOS y Windows Phone presentan diferentes alternativas para realizar la selección múltiple de elementos en una lista.

lista en su modo de edición, lo que permite seleccionar varios ítems a la vez.

Cuadros de diálogo

Hay casos puntuales en los que hay que interrumpir al usuario de forma temporal para que tome una decisión o para explicarle mejor algo que ha sucedido antes de continuar una tarea. Mientras los diálogos están visibles en pantalla no es posible hacer otra cosa en el resto de la aplicación.

Cuando se trata de cuadros que contienen avisos que no requieren una toma de decisión, estos son informativos y tienen únicamente un botón que se ocupa de cerrarlos. Es recomendable limitar su uso para mensajes graves o trascendentales que no puedan esperar.

En otros casos, los cuadros de diálogo se usan para comunicar al usuario que debe tomar una decisión para poder continuar y puede elegir entre dos o más opciones disponibles.







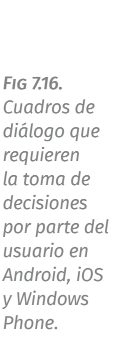


Fig 7.16. Cuadros de diálogo que requieren la toma de decisiones por parte del usuario en Android, iOS y Windows Phone.

Android usa extensamente los cuadros de diálogo. Las solicitudes pueden ser tan simples como «aceptar» o «cancelar», hasta diseños complejos con un formulario dentro.

En iOS se ubican en el centro de la pantalla, la mayoría de las veces con uno o dos botones en la zona inferior. En caso de que tengan más de dos botones, se muestran apilados uno sobre otro. Estos diálogos suelen ser tan simples que solo tienen un título, a veces acompañado de una descripción corta.

En Windows Phone los cuadros de diálogo se colocan en la zona superior y pueden ocupar una porción o —en casos excepcionales— toda la pantalla.

Notificaciones dentro de la app

Cuestiones como: ¿Qué está haciendo la app? ¿Cómo saber que la acción ha funcionado? ¿Ya terminó o hay que hacer algo más?, seguramente pasan por la cabeza de un usuario cuando no tiene ninguna confirmación visual de que la acción que acaba de realizar ha ido bien.

Para mitigar esta incertidumbre, se aconseja mostrar explícitamente cómo han ido las cosas o que sucederá en breve con simples mensajes de confirmación. Este tipo de mensajes se presentan en pequeños avisos que desaparecen luego de unos segundos.

A diferencia de los cuadros de diálogo, las notificaciones no requieren la intervención del usuario ni tampoco interrumpen su flujo de trabajo.

Android incorpora las llamadas «tostadas» —*toasts* en inglés—. A nivel visual, es una pequeña «pastilla» en negro que se ubica en la zona inferior de la pantalla y por encima de cualquier otro elemento de la interfaz.

Esta notificación aparece solamente por un corto período de tiempo, con un texto —generalmente de una sola línea— que le da *feedback* al usuario, por ejemplo, mientras la app está guar-

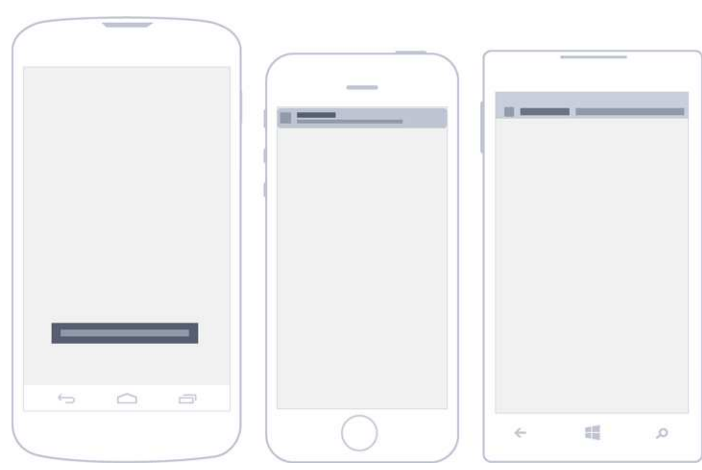


FIG 7.17. *Solo Android ofrece notificaciones in-app nativas, mientras que en iOS y Windows Phone deben programarse especialmente.*

dando un cambio. Al tratarse de un tipo de aviso que puede no ser percibido por el usuario, se usa para comunicar mensajes que no tienen una importancia crítica.

En contraste, en iOS y Windows Phone no existe una solución concreta similar a la propuesta de Android, por lo que las notificaciones dentro de la app quedan a cargo del diseñador, por ejemplo, usando librerías externas.

Introducción de datos

La introducción de datos en el móvil puede ser tediosa cuando se trata de campos que requieren el uso del teclado, un elemento que ocupa gran parte de la pantalla y que dificulta la navegación entre los campos para introducir información.

Los sistemas operativos han desarrollado teclados diferentes dependiendo del tipo de dato que deba ingresarse.

En lugar de teclados, es recomendable usar otras alternativas como menús deslizantes, desplegables, *checks* o cualquier opción similar a una lista donde el usuario pueda elegir entre varias opciones.







FIG 7.18. Diferentes tipos de teclados usados para introducción de datos en Android, iOS y Windows Phone.

Alternativamente, existen también componentes de hardware en el teléfono —como sensores de ubicación, cámaras y micrófonos— que pueden emplearse también para ingresar datos en la aplicación.

Un ejemplo de esto lo encontramos en Google, que ofrece la posibilidad de realizar consultas a través de voz en su buscador u obtener la ubicación en un mapa de acuerdo a la geolocalización, sin necesidad de escribir la posición geográfica donde nos encontramos.





FIG 7.19. El micrófono del teléfono se puede aprovechar para introducir datos sin necesidad de usar el teclado, como hace Google.

Gestos

Tocar es el *input* principal de los teléfonos móviles modernos. Todo queda en manos del usuario que manipula los elementos directamente en la pantalla. La acción y reacción suceden en el mismo sitio de manera similar a lo que ocurre en el mundo real.

Cuando se introdujeron las pantallas que soportaban múltiples gestos, parecía que las interfaces táctiles se iban a enriquecer considerablemente. La realidad es que, unos años más tarde, no se puede afirmar que los gestos más complejos hayan calado fuerte en los usuarios. Esto se debe, en parte, a que mientras más complicados son los gestos menos personas pueden realizarlos[6].

Por el contrario, gestos simples como tocar, arrastrar o deslizar —que requieren solo uno o dos dedos— han sido bien asimilados por los usuarios, que los encuentran naturales y familiares como los que realizan fuera de las aplicaciones.

El uso de los gestos se puede aprovechar en el diseño de aplicaciones, que debe considerarlos como medios para realizar acciones o navegar por los contenidos. Aquí también es importante aprovechar los conocimientos previos del usuario y ser consistentes con el sistema operativo.

Acciones básicas de la aplicación deberían poder llevarse a cabo mediante gestos simples para asegurarse que la mayor parte de los usuarios los pueden realizar, dejando aquellos más complejos como una forma alternativa de interactuar con la interfaz de la app.

Cada SO se ha esforzado en imponer sus propias convenciones[7], pero afortunadamente hay gestos compartidos entre Android, iOS y Windows Phone. A continuación, una lista de los más comunes y para qué se utilizan:

6 Fidalgo, Armando. Interfaces táctiles: el desafío de las tabletas. http://www.slideshare.net/Afidalgo/interfaces-tctiles-ux-spain

7 Wroblewski, Luke. Touch Gesture Reference Guide. http://lukew.com/touch

Tocar
Toca la superficie con la yema del dedo.



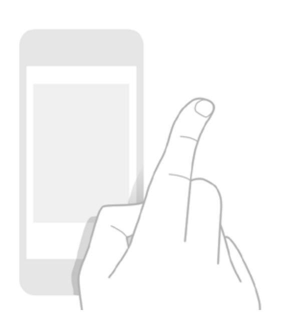
- *Selecciona la acción primaria.*
- *Selecciona la acción primaria.*
- *Selecciona la acción primaria.*

Arrastrar
Mueve un dedo sobre la superficie sin perder el contacto.



- *Archivar o borrar al arrastrar horizontalmente en listas.*
- *Mostrar un botón de eliminar al arrastrar horizontalmente en listas. Mover ítems en listas.*
- *Cambiar a otras pestañas o sectores de una vista Panorama.*

Deslizar
Desliza rápidamente y sin detenerte la yema de un dedo sobre la superficie.

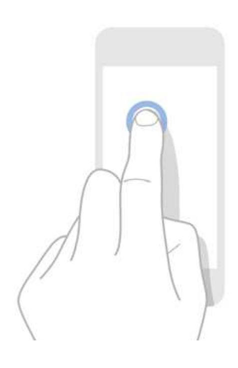


- *Desplazarse a través del contenido. Cambiar a otras pestañas.*
- *Desplazarse a través del contenido.*
- *Desplazarse a través del contenido. Cambiar a otras pestañas o sectores de una vista Panorama.*



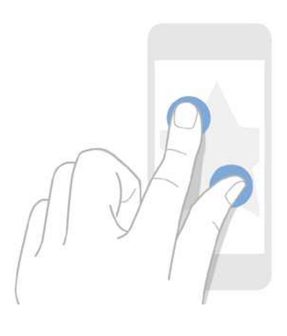
Mantener pulsado
Toca la superficie durante un período prolongado de tiempo sin mover el dedo.

- *Entrar al modo de edición de listas.*
- *Mostrar un tooltip. Aumentar el contenido visible bajo el dedo.*
- *Mostrar un tooltip sin seleccionar el elemento.*



Doble toque
Toca rápidamente la superficie dos veces con la yema del dedo.

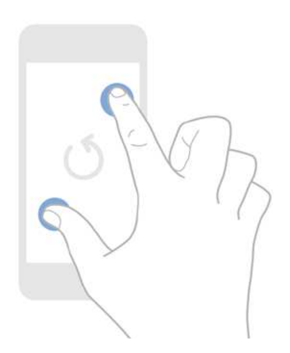
- *Alternar entre acercar y alejar. Seleccionar texto.*
- *Alternar entre acercar y alejar.*
- *Alternar entre acercar y alejar.*



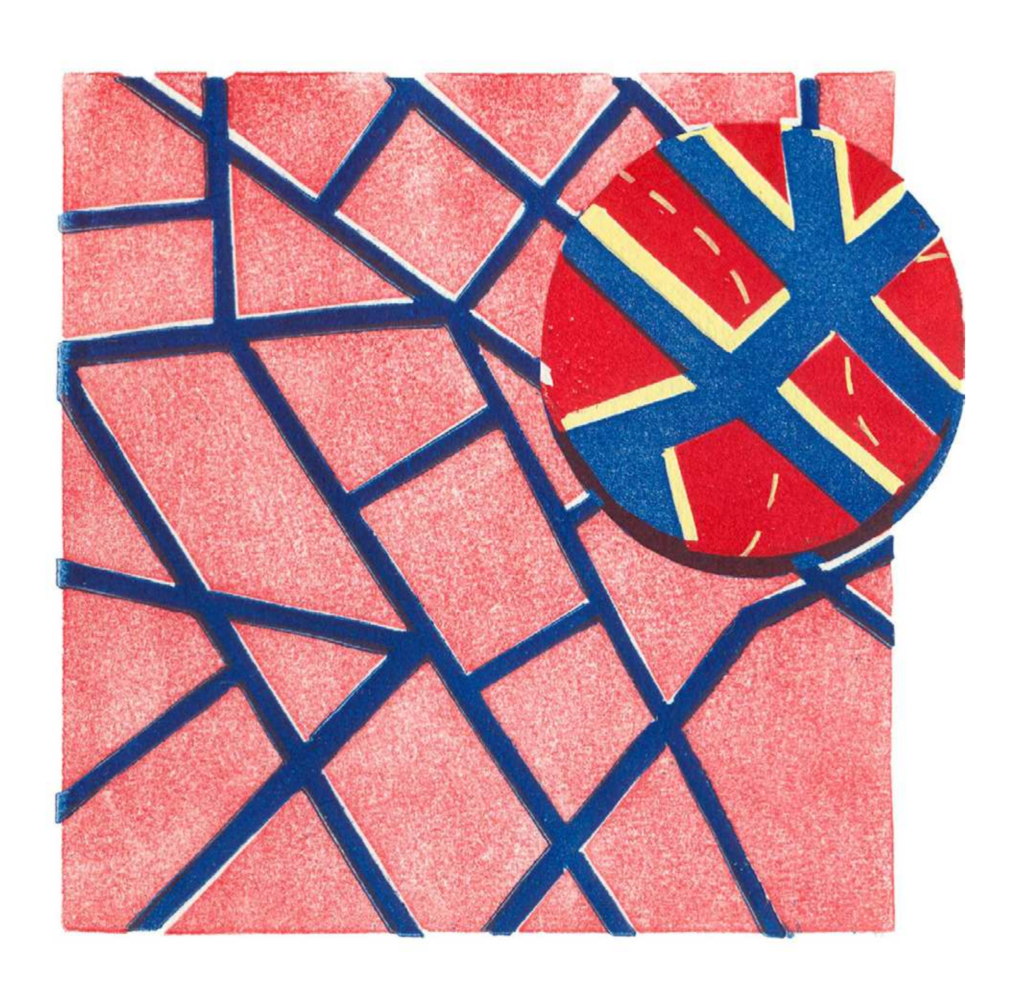
Juntar y separar
Toca la superficie con dos dedos y júntalos o sepáralos.

- *Aumentar o reducir el zoom.*
- *Aumentar o reducir el zoom.*
- *Aumentar o reducir el zoom.*

Girar
Toca la superficie con dos dedos y júntalos o sepáralos mientras giras la muñeca.



- *Girar una imagen o un mapa.*
- *Girar una imagen o un mapa.*
- *Girar una imagen o un mapa.*



8

Diseño visual

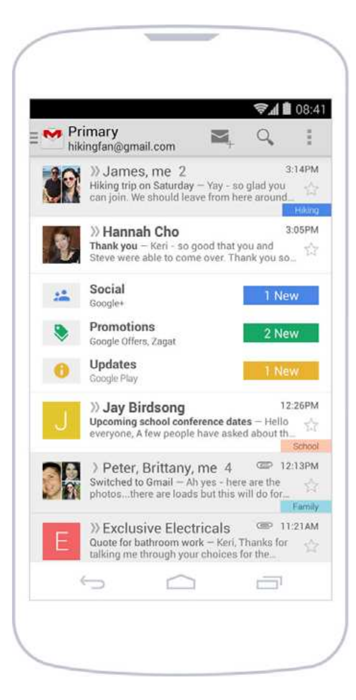
El diseño visual es para muchos diseñadores la etapa más divertida del proceso. Aquí se da vida a los wireframes con un estilo que está marcado tanto por el diseñador como por la personalidad de cada sistema operativo.

El estilo de las interfaces

La interfaz de una aplicación es como la ropa que viste para salir a la calle. Es también la capa que hay entre el usuario y el corazón funcional de la app, el lugar donde nacen las interacciones.

En mayor medida está compuesta por botones, gráficos, íconos y fondos, que tienen una apariencia visual diferente en cada uno de los sistemas operativos, porque Android, iOS y Windows Phone tienen su propia forma de entender el diseño.

El trabajo del diseñador consiste en interpretar la personalidad de cada sistema operativo, aportando su propia visión y estilo de diseño, para conseguir aplicaciones que, además de ser fáciles de usar, sean distintas a las demás y tengan coherencia visual con la plataforma que las acoge.



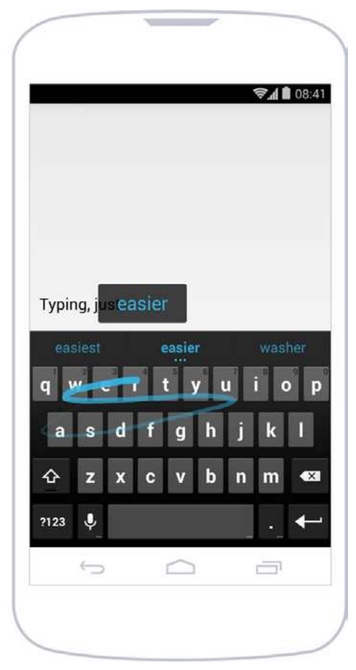


Fig 8.1. Android tiene un estilo de diseño simple basado en Roboto como fuente principal.

La belleza simple de Android

El diseño en Android está basado en una pulcritud brillante en la composición de la interfaz. Cada gráfico, botón y texto está acompañado por la idea de limpieza visual pero, a la vez, deslumbra con pequeños detalles.

Roboto, la tipografía propia de este sistema operativo, es en gran parte su seña de identidad y se combina con un estilo de botones y colores bien definido. Android se apoya en la simplicidad, controlada pero no aburrida, que, en ocasiones, rompe o trasciende sus propios formalismos para encantar al usuario.

iPhone, en busca de la ligereza visual

Dejando atrás un estilo que lo marcó hasta su versión 6, iOS defiende actualmente una ideología compartida también con

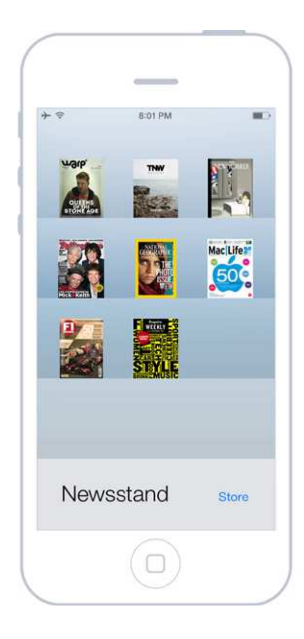




FIG 8.2. La claridad visual, conseguida por la elección cromática y de tipografía de iOS, intenta jerarquizar el contenido sobre fondos y elementos secundarios.

otros sistemas operativos: despojarse de elementos innecesarios, privilegiando el contenido por sobre el contenedor.

Para conseguir esto reduce controles y gráficos a su mínima expresión, siempre con la idea de aligerar visualmente los elementos en pantalla. Este concepto se respalda con una elección tipográfica —Neue Helvética, muchas veces en sus variables más *light*— y la elección de colores —blancos para fondos y fuertes para íconos y textos—.

Cada componente de la interfaz se trata como una capa superpuesta a la otra, a veces con cierto grado de transparencia y difuminado, transmitiendo una sensación de continuidad y permanencia en el contexto.





FIG 8.3. El diseño de Windows Phone tiene con Metro un estilo plano, muy basado en su retícula y tipografía, evocando la escuela suiza de diseño.

El diseño plano de Windows Phone

El estilo de diseño de Windows Phone se basa en un diseño plano, despojado de relieves, degradados y decoraciones estéticas excesivas. Esta limpieza visual también se aplica a los contenidos, donde solo aquellos importantes permanecen en pantalla, luciéndose en su contexto.

La interfaz consiste en una aproximación infográfica para sus íconos, con un marcado uso de la retícula y la tipografía como uno de los principales recursos para dotar de personalidad al diseño.

Interfaces nativas o personalizadas

Las interfaces nativas se basan en elementos —botones, listas y encabezados— que vienen preestablecidos en cada plataforma. Tienen un aspecto ya definido en cuanto a las características básicas de su apariencia como color, tamaño o tipo de fuente, que pueden ajustarse en mayor o menor medida para que se correspondan con la estética buscada.

En el momento de comenzar a diseñar es recomendable definir la interfaz con elementos nativos. De esta forma, se consigue una buena base sobre la cual trabajar y no es necesario crear todos los elementos desde cero.

El inconveniente de las interfaces nativas es que limitan la personalidad del diseño y, en algunos casos, es necesario ir un paso más allá. En situaciones como esta, todos o algunos elementos de la interfaz pueden ser personalizados, lo cual se logra creándolos de nuevo como imágenes. Por ejemplo, un elemento visual personalizado podría ser un campo de entrada de texto de un formulario que se genera como imagen, para aprovechar la posibilidad de incluir texturas, relieves o sombras específicas, que un elemento nativo no ofrece.

Diseñar una interfaz personalizada tiene que planearse de antemano porque representa una mayor complejidad y tiempo de desarrollo. De la misma manera, no siempre los diseños de este tipo de interfaces se trasladan a la aplicación funcional de forma fidedigna, pues su correcta implementación queda en manos de la pericia del desarrollador.

¿Cuándo usar una u otra interfaz?

En la mayoría de los casos no se trata de elegir entre una u otra, sino de alcanzar el balance adecuado combinando ambas. La



FIG 8.4.
La app de Wunderlist incorpora elementos personalizados en su interfaz —como el campo de texto— bien integrados.

situación más habitual suele ser partir de una interfaz nativa y personalizar solo aquellos elementos que se consideren necesarios.

El tipo de aplicación también tiene incidencia en este asunto. Por ejemplo, las aplicaciones que dan valor a los detalles visuales y a la experiencia en general, suelen contar con más de un elemento personalizado. Tal es el caso es Path y su forma de añadir «momentos», algo diseñado y desarrollado a medida.

Por el contrario, hay otras aplicaciones que dan más valor al cumplimiento de tareas y que se benefician de tener una apariencia más limpia que no distraiga del proceso. Un ejemplo de ellas es Whatsapp, cuyo aspecto visual está casi completamente basado en elementos nativos del sistema operativo.

Además del objetivo o del tipo de aplicación, hay otras variables para considerar. Las interfaces nativas tienen un punto a favor y es que están constituidas por elementos que el usuario ya conoce y a los que está habituado, por lo tanto, no representan un nuevo aprendizaje. Esto puede incidir favorablemente en la usabilidad de la app.

Por otro lado, las interfaces personalizadas pueden ofrecer una apariencia más acabada, pero al trabajar con ellas hay que considerar la compatibilidad con múltiples dispositivos —cómo se visualiza el mismo diseño en teléfonos diferentes— y el rendimiento general —un uso excesivo de imágenes puede hacer que la app sea más lenta—.

La identidad visual

Una aplicación es, entre otras cosas, una pieza de comunicación. Forma parte de un sistema y es una oportunidad para extender la identidad de una empresa o producto.

A través de las diferentes pantallas de la app, los colores, tipografías y fondos actúan como elementos que reflejan esa identidad.

Claramente, uno de los componentes de la identidad es la marca. Aunque sea tentador hacer un uso extensivo y repetido de ella, se recomienda incluirla en lugares propicios para tal fin, como pantallas introductoras, pantallas para ingresar clave y usuario, o en la sección «Acerca de». De esta forma, se puede asegurar una correcta exhibición de la identidad sin afectar la navegación y la experiencia de uso.

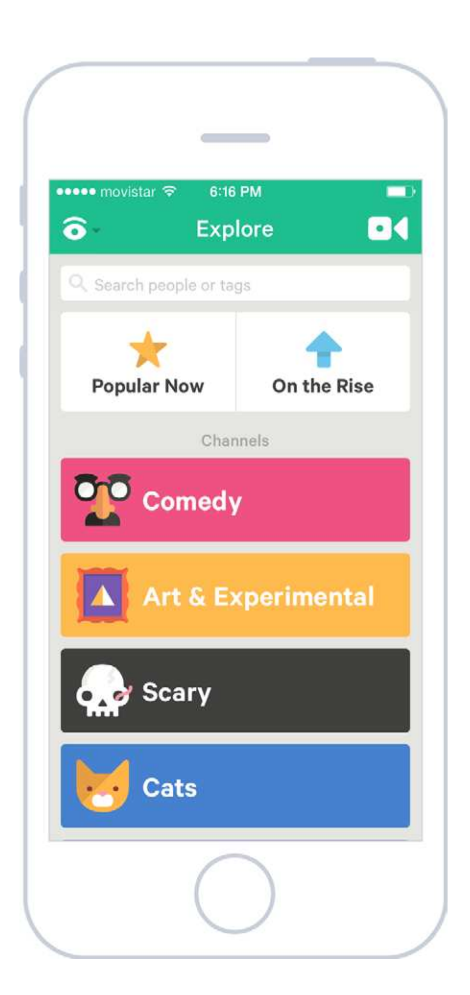


FIG 8.5. Vine combina su color corporativo en la cabecera con otros colores en los hashtags.

ÍCONOS Y PANTALLA INICIAL

Dicen que la primera impresión es la que cuenta. En el mundo de las aplicaciones esa primera impresión está limitada a dos componentes visuales: el ícono de lanzamiento y la pantalla inicial —también llamada *splash*— que se mostrará muchas veces al abrir la aplicación.

Estos elementos se verán antes que nada, incluso, antes de empezar a usar realmente la aplicación. No menospreciar su importancia y darles la atención que merecen, garantiza arrancar con el pie derecho.

Ícono de lanzamiento

Hay que pensar en la aplicación como un producto que estará en un escaparate junto a muchos otros y el ícono de lanzamiento es el *packaging* que lo envuelve.

En primer lugar, este ícono servirá para representar a la app en las diferentes tiendas de aplicaciones —junto a las pantallas y textos promocionales— como elemento de venta para convencer al usuario de descargarla.

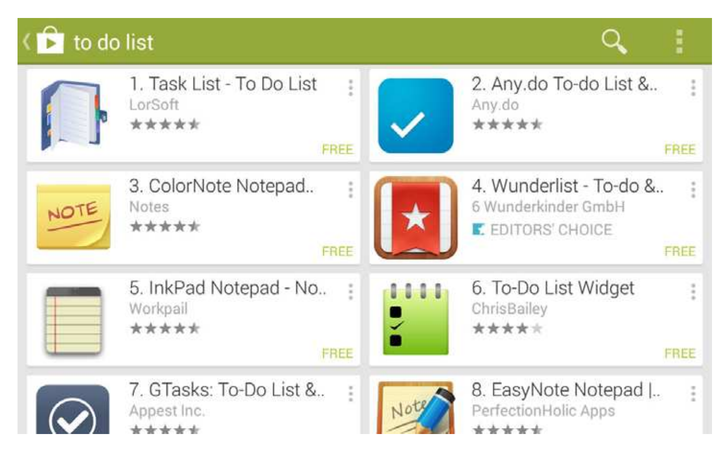


FIG 8.6. *Una aplicación compite con muchas otras cuando un usuario llega a ella a través de los resultados de búsqueda, en este caso Google Play.*



FIG 8.7.
El ícono de lanzamiento de una app debe tener personalidad para diferenciarla de las demás en una tienda, pero también representar claramente lo que hace. En el ejemplo, hay íconos de diferentes apps de tareas y una red social, ¿puedes descubrir cuál es?

Sorteado este paso, y una vez instalada en el teléfono, la aplicación convivirá con muchas otras que el usuario haya instalado; por eso, el ícono de lanzamiento debe ser distintivo y representativo de la app. Distintivo, porque tiene que separarla de las demás, incluso de aquellas que cumplan funciones similares, y representativo, porque sus características visuales tienen que comunicar claramente el objetivo principal de la aplicación. Las formas simples, no muy cargadas y cuidadas en sus detalles, suelen ser las que tienen mayor efectividad.

El tamaño también es algo a tener en cuenta. Algunas veces, el ícono se verá realmente grande, por ejemplo en las tiendas de aplicaciones, pero otras, como cuando la app está instalada, se verá mucho más pequeño. Al diseñarlo hay que considerar todas las posibilidades, añadiendo más o menos detalle a la imagen en función de sus dimensiones.

Siendo aún más específicos, cada uno de los SO tiene diferentes requisitos que debe cumplir el ícono de lanzamiento. Tanto Android[1] como iOS[2] ofrecen detalladas guías del estilo visual y características técnicas que debe tener esta imagen.

1 http://developer.android.com/design/style/iconography.html

2 http://developer.apple.com/library/ios/#documentation/userexperience/conceptual/mobilehig/IconsImages/IconsImages.html





FIG 8.8. *En Android los íconos suelen tener sombras y juegan con una ligera perspectiva.*

En Android los íconos son objetos representados frontalmente con una ligera perspectiva, como si fueran vistos desde arriba. Dan sensación de volumen y profundidad, jugando con transparencias para integrarse mejor a la pantalla. Las formas son distintivas y con medido realismo.






FIG 8.9. *Íconos encerrados en formas contenedoras con esquinas redondeadas, el sello de identidad de iOS.*

La situación es diferente en iOS: los íconos son por lo general representaciones muy simplificadas de objetos reales, o abstracciones del concepto de la app. Por lo general el ícono contiene un solo elemento principal, sin muchos detalles, que reposa sobre un fondo opaco. Para completar la apariencia visual que el ícono tiene en la pantalla del teléfono, iOS añade bordes redondeados a la imagen, que debe ser cuadrada y cubrir la totalidad del tamaño requerido en cada caso.






FIG 8.10. *Aunque hay excepciones, en Windows Phone los íconos de lanzamiento deben ser formas simples y blancas.*

Por su parte, en Windows Phone tienen un estilo bastante característico: los íconos son pictogramas. Formas extraordi-

nariamente simples y de colores planos —sobre todo blanco—, casi sin detalles e integradas perfectamente a su contenedor. En este caso, la transparencia es fundamental ya que las imágenes están ubicadas dentro de una forma cuadrada, que puede cambiar el color de fondo de acuerdo a las preferencias cromáticas del usuario.

Íconos interiores

Ya dentro de la aplicación, los íconos interiores tienen un papel menos estelar y más funcional que los de lanzamiento. Puede ser que pasen desapercibidos, pero su trabajo es importante y silencioso y, como tal, digno de tener en cuenta.

Su uso suele estar asociado a tres escenarios. El primero, como ayuda visual para reforzar información, por ejemplo, en un cuadro de diálogo con una alerta. En el segundo caso, los íconos interiores actúan como complemento de elementos interactivos, como cuando se encuentran dentro de botones o pestañas. Finalmente, su función puede ser la de mejorar la utilización del espacio, en este caso, el ícono resume visualmente algo que en forma de texto sería muy extenso o complejo de entender.

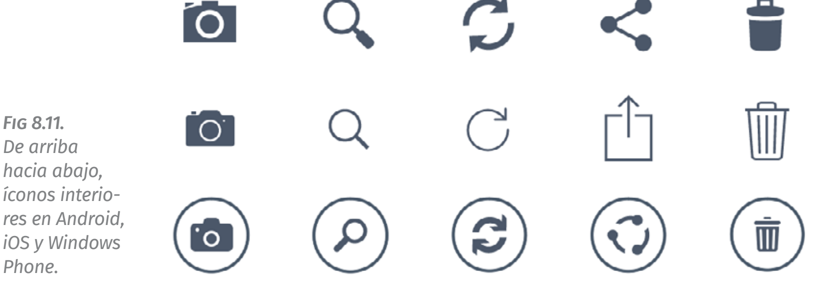


FIG 8.11. De arriba hacia abajo, íconos interiores en Android, iOS y Windows Phone.

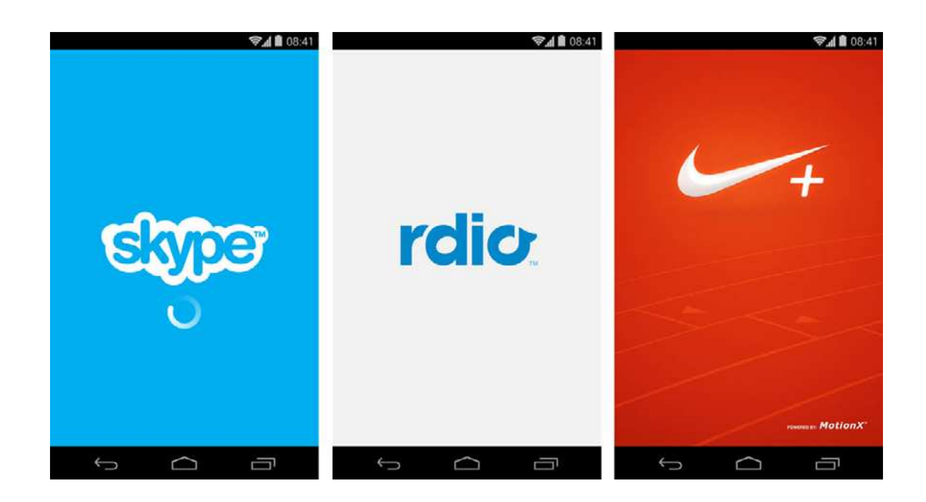


FIG 8.12. Las pantallas iniciales en Android contienen poca información y solo se muestran brevemente durante la carga.

Los íconos tienen que transmitir por sí solos la acción que ejecutan y esto depende del contexto. Por ejemplo, un ícono de «eliminar» puede referirse a un solo elemento en particular o a varios, dependiendo de dónde esté ubicado y a qué elemento de la interfaz esté visualmente asociado.

Cuando los íconos acompañan determinadas acciones —si no tienen etiquetas de texto que ayuden a ejemplificar su función— se vuelve más importante que sean claros y representativos. Esto ocurre cuando, por limitaciones de espacio, no puede incluirse un ícono y texto al mismo tiempo. Además, es una tendencia que está empezando a verse cada vez más en las aplicaciones.

La interpretación de íconos supone cierta subjetividad que hay que tratar de eliminar haciendo un uso correcto de ellos. Por ejemplo, en cada plataforma algunas acciones como «buscar», «guardar» o «editar» tienen íconos asociados. El usuario ya los conoce y sabe qué significan, por lo tanto, darles el uso que se espera de ellos ayudará a alcanzar consistencia y mejorará la usabilidad de la aplicación.

Pantalla inicial

Conocida también como *splash*, es la primera pantalla que verá el usuario al iniciar la app. Su uso está siendo cada vez más limitado —y evitado—, por lo que generalmente se muestra rápidamente la primera vez que se abre la aplicación. Esta pantalla sirve como presentación del contenido mientras se realiza la carga inicial, por tanto, es normal que se incluya un elemento indicativo de carga junto a los demás elementos gráficos.

La pantalla inicial tiene un carácter tan efímero que pocas veces se ve más de un par de segundos. Debido al poco tiempo de vida que tiene, la información que muestra se debería limitar al nombre y versión de la aplicación, y poco más, ya que desaparecerá tan pronto como la aplicación termine de cargarse.

En algunos casos, este splash se trata de una representación del contenedor de la aplicación: una imagen casi idéntica a la que verá el usuario cuando la carga haya finalizado, pero sin

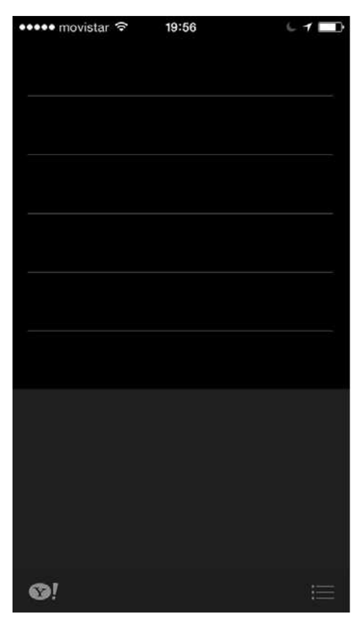




FIG 8.13. *La recomendación en iOS es usar una pantalla inicial que sea una representación del contenedor, vacío de información.*

aquellos datos que pueden llegar a cambiar como textos, etiquetas en botones y elementos de la barra de estado, haciendo que, aparentemente, la carga resulte más fluida[3].

En el caso de Windows Phone, el sistema operativo se encarga de gestionar la pantalla de inicio que consiste en una ampliación del ícono de lanzamiento.

En cuanto a la orientación de esta pantalla en teléfonos móviles, hay que recordar que suele mostrarse de forma vertical, mientras que en las tabletas, es necesario determinar la orientación que se está usando al momento de mostrar el splash y, en función de eso, hacer uso de la versión correspondiente al modo horizontal o vertical.

Grilla o retícula de construcción

La grilla o retícula —*grid* en inglés— es la estructura invisible sobre la cual se apoyan todos los elementos visuales. Su función es la de separar cada uno de los componentes de la interfaz en un espacio ordenado, organizando los sitios que quedarán en blanco y aquellos que contendrán formas. Una retícula bien definida se transforma en una ayuda al diseño que, generando orden y simplicidad, mejora la usabilidad de la app.

En su forma más básica consta de un módulo base: un cuadrado de un tamaño determinado que se usa como medida de referencia. A su vez, este módulo puede dividirse en submúltiplos para espaciados más pequeños.

Mientras el diseño de la interfaz está en desarrollo, la retícula se representa por medio de líneas guía. Una vez terminada la estructura, se puede percibir por el llamado «ritmo visual» que ubica los elementos armónicamente en el espacio.

En el diseño de interfaces para móviles, la grilla permite establecer márgenes y determinar la ubicación de los botones, la

3 http://blog.manbolo.com/2012/06/27/of-splash-screens-and-ios-apps

separación de la tipografía y el espacio interior y exterior de los contenedores. Por supuesto, cada uno de los sistemas operativos tiene diferentes retículas y por lo tanto, distintos módulos.

Android

En Android, el módulo base es de 48dp que equivale aproximadamente a nueve milímetros, tamaño mínimo recomendado para elementos interactivos. Basarse en este tamaño y respetar estas dimensiones para los botones, permite asegurar que estos podrán ser tocados con el dedo sin problemas, cuestión fundamental en el diseño para móviles.

Para el espaciado y separación, en cambio, se usa un módulo de 8dp. Por ejemplo, el contenido de filas tiene una separación —superior e inferior— de 4dp, lo que hace que cuando dos filas están una sobre otra, se sumen conformando un espacio total de 8dp entre ellas. En los márgenes laterales, los diseños suelen tener 16dp o, dicho de otra forma, dos módulos de 8dp juntos[4].

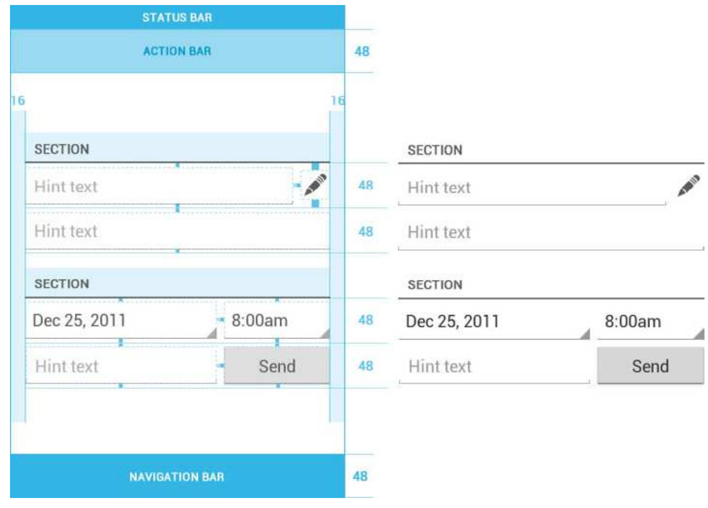


FIG 8.14. En la imagen se puede ver claramente cómo una interfaz en Android está compuesta por un módulo base de 48dp.

4 http://developer.android.com/design/style/metrics-grids.html



FIG 8.15. En iOS el módulo base es de 44px.

iPhone

Los diseños para iPhone también se basan en una retícula, solo que en este caso el tamaño del módulo base es de `44px`, asegurando que botones y elementos en listas puedan ser pulsados sin problemas. En ese ritmo se basan muchas aplicaciones diseñadas para iPhone y es el mismo que recomienda Apple.

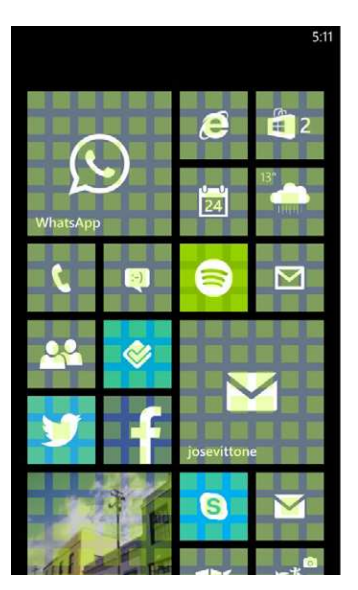
Este módulo de `44px` se divide en otros de `11px` que, repetidos la cantidad de veces deseada, crean los espacios y separación entre tablas, botones y otros elementos de la interfaz, generando un ritmo vertical[5].

Windows Phone

En Windows Phone la retícula se hace más evidente debido al uso que hace de formas cuadradas en la interfaz. La pantalla principal del sistema operativo está basada en los llamados azulejos o *tiles*, que permiten apreciar claramente la distribución y tamaño de los elementos principales.

5 http://aentan.com/design/new-visual-proportions-for-the-ios-user-interface/

Fig 8.16.
En Windows Phone la retícula es un pilar fundamental sobre el cual se apoyan todos los elementos.





En este caso, la retícula se apoya en un módulo de `25px` con separaciones de `12px`. Esta fórmula, repetida por toda la pantalla, crea una estructura visual sobre la cual se crean los diseños. Adicionalmente, las filas y columnas pueden agruparse de distintas maneras para lograr diseños personalizados que hacen diferentes usos del espacio.

Listas, gráficos, miniaturas de fotos y botones, toman como base esta retícula, logrando una apariencia estable y dando sensación de orden, simplicidad y limpieza visual, a través de todas las pantallas de una app de Windows Phone.

Tipografía

Como en cualquier diseño, el objetivo de la tipografía es conseguir que el texto se lea con claridad. Esto se logra no solo con una adecuada elección de la fuente, sino también gestionando su tamaño, separación entre líneas, ancho de columnas y contraste visual con el fondo.

Este último punto, el contraste, es más importante de lo que puede creerse a simple vista. Un móvil es un dispositivo que muchas veces se usará fuera de casa, por ejemplo, en la calle. En algunos momentos el sol dará directamente sobre la pantalla y si no hay un buen contraste entre tipografía y fondo, la información en pantalla será imposible de leer.

La tipografía es un componente que, como botones y gráficos, también se asienta en una retícula que definirá su ubicación y posición dentro del contexto general de la pantalla.

Serif o sans-serif

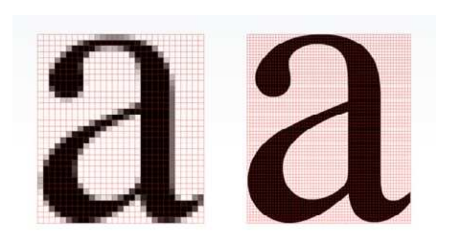
Generalmente el debate sobre la elección de la fuente suele concentrarse mayormente entre las tipografías que tienen serifa y las que no. Si bien es cierto que en tamaños pequeños y bajas resoluciones rinden mejor aquellas fuentes más limpias, abiertas y *sans-serif,* también pueden considerarse las *serif* para los títulos principales, cuando cuenten con un tamaño en el que las serifas no sean un impedimento para la lectura.

Aunque en móviles no sea habitual una lectura de texto prolongada, la correcta legibilidad es una parte fundamental del diseño; por esta razón, la tipografía es tan importante como cualquier otro elemento visual que se incorpore en una interfaz y su elección no debería descuidarse.

Legibilidad y resolución

Por ser un soporte digital, los móviles tienen características propias y algunas limitaciones ajenas a los medios impresos tradicionales. La pantalla influye bastante en el comportamiento y desempeño tipográfico si se tiene en cuenta que, en algunos casos, es sumamente pequeña.

Fig 8.18.
En terminales con buena calidad de pantalla los caracteres son más legibles.



Actualmente, muchos dispositivos tienen pantallas con buena resolución, lo que elimina un factor de complejidad y una gran preocupación a la hora de elegir qué fuente se usará. Este tipo de pantallas suele encontrarse en móviles de alta gama, sin embargo, hay otros teléfonos con características no tan interesantes.

En este último caso, elegir la tipografía es una tarea bastante difícil que no se limita a la elección de familia y tamaño, ya que, mientras más pequeña sea la pantalla, hay más posibilidades de que la forma en que se pintan en ella los caracteres, sea más pobre.

Tamaños mínimos

En soportes digitales, como un móvil o una tableta, algo que condiciona el tamaño tipográfico es la distancia a la cual se

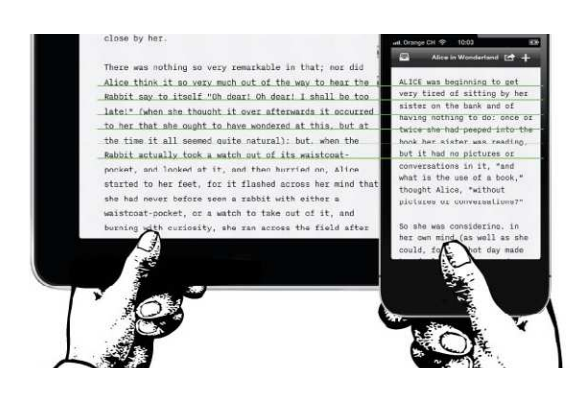


Fig 8.17.
El tamaño de pantalla y la distancia de lectura condicionan la elección de los tamaños de la fuente.

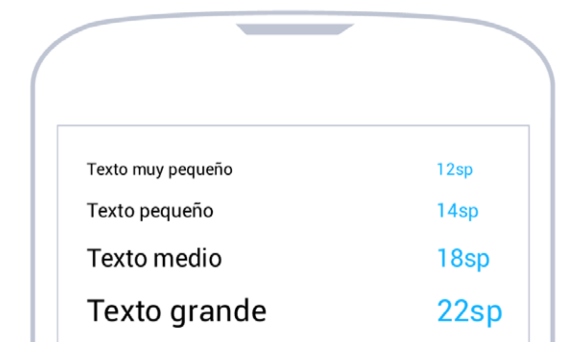


FIG 8.19. Diferentes tamaños de fuentes en Android de acuerdo al uso y jerarquía de los elementos.

sujeta el dispositivo. Los móviles suelen sostenerse de forma que la pantalla está más cerca del ojo del lector que en el caso de una tableta, lo que permite que el tamaño de la tipografía sea más pequeño.

Al mismo tiempo, en los teléfonos el espacio en pantalla es mucho menor, lo cual obliga a ajustar el interlineado y la separación entre caracteres, para aprovechar el área disponible sin perjudicar la lectura.

Los tamaños mínimos pueden variar dependiendo del sistema operativo, la resolución de la pantalla y la fuente que se elija. En todo caso, cuando se trata de tamaños pequeños, se aconseja elegir fuentes de formas simples y abiertas, con espaciado entre caracteres, líneas y márgenes para dar aire visual que facilite la lectura.

La mejor forma de asegurar una correcta legibilidad no es otra que ponerla a prueba en el teléfono para el cual se diseña. Los diseños en pantalla de ordenador suelen ser engañosos y comprobarlos en el entorno más real posible, permite realizar ajustes y correcciones hasta conseguir el tamaño adecuado.

En Android, el tamaño tipográfico se mide en sp —*scaled pixels* o píxeles escalados—, una forma de modificar la escala de las fuentes de acuerdo al tamaño de pantalla y a las preferencias definidas por el usuario en su configuración del teléfono. Los tamaños más comunes van desde `12sp` hasta `22sp`.

El tamaño es variable en iOS, dependiendo de dónde está ubicado el texto. Teniendo en cuenta tamaños para la densidad «retina», los títulos principales rondan los `34px`. Por otro lado, el tamaño en etiquetas dentro de botones importantes es aproximadamente `28px`. A partir de allí, va disminuyendo en los diferentes elementos hasta llegar al tamaño más pequeño, cercano a los `14px`. Sin embargo, se recomienda no usar dimensiones inferiores a `20px` en los textos de lectura.

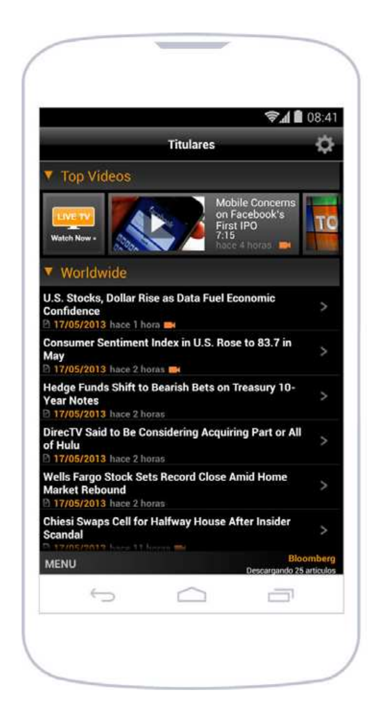
En Windows Phone, más que otras plataformas, el diseño recae en la tipografía. Una mala elección del tamaño de la fuente en una app que cuenta con pocos elementos visuales, puede llevarla a la ruina. Las recomendaciones son: no usar un tamaño menor a `20px` para los textos más pequeños y en el caso de títulos, considerar tamaños que pueden llegar, incluso, a `70px`.

Jerarquías

Como elemento visual y componente de una interfaz, la tipografía también es susceptible de ser jerarquizada. Su importancia depende no solo de la función que cumple, sino también de la información que contiene y su posición en pantalla.

Para definir diferentes niveles de protagonismo en un texto se puede apelar, además del tamaño, a las variantes —negrita, regular o *light*, por ejemplo— y al color.

Un texto con mayor jerarquía sería aquel que se ubica como título principal de sección. Por otro lado, en la información contenida dentro de una fila en una lista, puede haber varias jerarquías, por ejemplo, en el caso de un correo, podría verse el nombre de la persona que lo envía, un resumen del contenido y la fecha de envío; todos estos elementos tienen diferente importancia y definirla es el primer paso para saber qué estilo tipográfico aplicar.



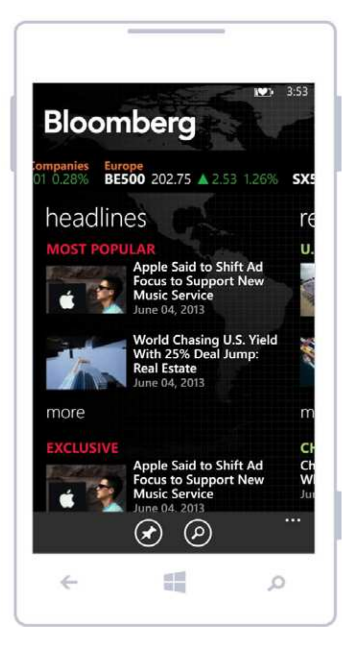


FIG 8.20. Mientras que en la pantalla de la izquierda no se aprecia una jerarquía clara, a la derecha es más evidente la importancia de cada elemento de acuerdo al tamaño de la tipografía.

Las tipografías de cada sistema operativo

Tanto Android, como iOS y Windows Phone, tienen sus preferencias tipográficas y su propio set de fuentes de sistema. Esto no quiere decir que el diseñador deba apegarse necesariamente a esta selección, pero elegir alguna de las opciones disponibles ayudará a vincularse con la identidad de cada plataforma.

En el caso de Android, la familia Droid Sans fue una de las más utilizadas y marcó una época dentro de esa plataforma. Sin embargo, en las últimas versiones de este sistema operativo ha sido reemplazada por Roboto, especialmente diseñada para móviles de alta resolución y con variantes que van desde una extra delgada hasta una *black*[6].

6 http://developer.android.com/design/style/typography.html

Fig 8.21. *Neue Helvética es un clásico de iOS, a pesar de los proble-mas que pue-de presentar en tamaños reducidos.*

Neue Helvética Light

ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz

Neue Helvética Medium

ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz

Fig 8.22. *Roboto es la fuente emblema de Android, pensada con variables para móviles con buena calidad de pantalla.*

Roboto Regular

ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz

Roboto Bold

ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz

Fig 8.23. *Segoe UI es la fuente característica en Windows Phone. En sus interfaces tiene un gran protagonismo.*

Segoe UI SemiLight

ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz

Segoe UI Semibold

ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz

La fuente emblemática de iOS en los últimos años ha sido Neue Helvética y es normal encontrar muchas aplicaciones que la usan. Con el paso del tiempo se ha transformado casi en una marca de identidad de este sistema operativo; no obstante, hay más de 260 fuentes disponibles que pueden usarse de forma nativa[7].

Windows Phone introdujo Segoe UI como un sello que acompaña el estilo limpio, moderno y geométrico de su interfaz. Aun así, cuenta con una serie de familias tipográficas que la acompañan, sobre todo en casos que requieren caracteres especiales que no se encuentran en Segoe UI, como aquellos de idiomas extranjeros[8].

Cada uno de estos sistemas operativos también permite incorporar en el diseño fuentes adicionales a las predefinidas. Aunque esto sea posible, es importante recordar que variedad y calidad son dos cosas diferentes: muchas de las fuentes que pueden encontrarse, no han sido pensadas ni preparadas especialmente para una buena legibilidad en pantalla.

Color

El color es un recurso vital en el diseño de una aplicación. Su uso abarca encabezados, textos, botones, fondos y muchos otros elementos que conforman la interfaz. En algunas ocasiones, está asociado a la identidad —color corporativo— y en otras, responde a criterios estéticos y decisiones de diseño.

Un color por sí solo, salvo en el caso de colores reservados, no indica mucho. Como parte de un sistema cromático, el uso consistente, consciente y vinculado al contexto donde se aplica, es lo que lo llena de significado para el usuario.

7 iOS 7: Font List. http://support.apple.com/kb/HT5878

8 Text and fonts for Windows Phone. http://msdn.microsoft.com/en-us/library/windowsphone/develop/cc189010(v=vs.105).aspx





FIG 8.24. *Colores como el amarillo y rojo suelen reservarse para reforzar el significado y la importancia de algunos elementos visuales.*

Colores reservados

Hay ciertos colores que deben emplearse de forma cuidadosa porque tienen connotaciones que no pueden obviarse. Se llaman justamente «colores reservados» porque su uso debería limitarse especialmente a los nombrados a continuación:

- *Rojo:* Para errores y alertas importantes. Es un color que naturalmente indica peligro y llama la atención para centrarse inmediatamente en lo que está ocurriendo.
- *Amarillo:* Prevención. Señala que la acción que va a realizarse implica la toma de una decisión que ocasiona alguna consecuencia, por lo cual hay que estar alerta.
- *Verde:* Mensajes de éxito y confirmación de que una acción se ha realizado correctamente.

En textos

El color se puede usar para destacar aquellas frases o palabras que pueden ser pulsadas, como enlaces. En este caso, es importante mantener la consistencia visual para permitir al usuario, intuitivamente y a simple vista, saber cuáles son los elementos tipográficos interactivos.

Otra función que cumple el color al ser usado en textos es la de jerarquizar el contenido. La información complementaria puede ser destacada o minimizada dependiendo del color que se

use; por ejemplo, aquella que reviste cierta importancia podría destacarse en el texto con un color diferente. De la misma forma, el color también puede usarse para identificar información secundaria, usando tonos más claros sin tanto protagonismo.

En fondos

En el caso del color de fondo, este debe estar en consonancia con el elegido para la tipografía, ya que es necesario un contraste mínimo por cuestiones de legibilidad y accesibilidad.

Hay que tener en cuenta que los fondos oscuros suelen cansar la vista más rápidamente, por lo tanto, si la app es de uso frecuente o requiere pasar cierto tiempo leyendo, es conveniente revisar la elección cromática y llevar el color de fondo hacia alternativas más claras. Sin embargo, los colores oscuros en el fondo sí pueden ser una buena alternativa cuando el contenido sea muy visual, como fotografías o vídeos, ya que ayuda a resaltar estos elementos.

En elementos interactivos

El color puede utilizarse como respuesta o *feedback* a acciones concretas del usuario, un uso que muchas veces no se tiene en cuenta. Para ilustrar con un ejemplo, los elementos seleccionados o pulsados, como botones o filas, pueden destacarse con un color que indique visualmente dónde se ha pulsado, lo cual suele ser particularmente difícil de saber en un móvil.

En el caso de elementos deshabilitados, generalmente el color es más claro que cuando están en su estado normal, incluso, puede apelarse al uso de transparencias. De cualquier forma, el objetivo es indicar de una forma evidente que ese elemento no producirá ningún efecto al ser pulsado.

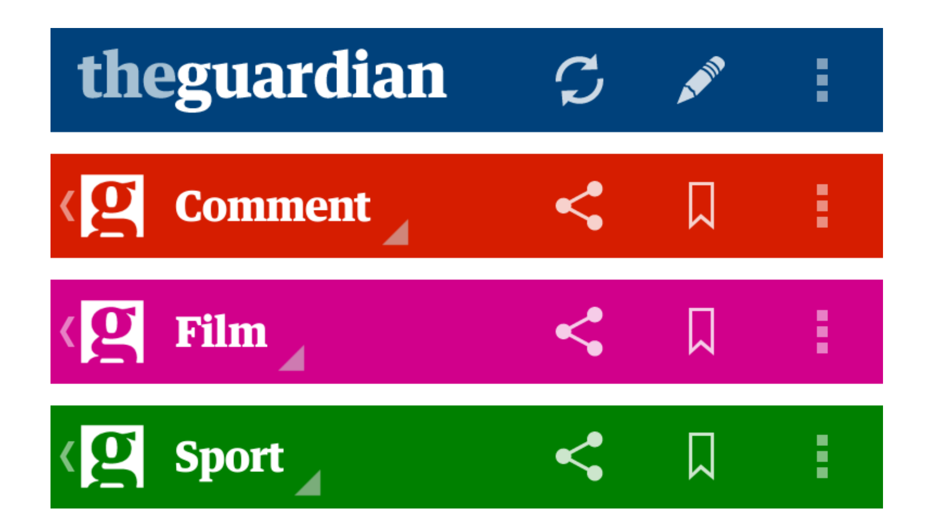


FIG 8.25. La aplicación de «The Guardian» usa el color en los encabezados para diferenciar cada una de las secciones principales.

En encabezados

Además de destacarlos como elemento principal, cuando el color se usa en encabezados tiene que armonizar completamente con el fondo y los otros elementos de la pantalla. A fin de cuentas, se trata de un importante espacio que se usará, por ejemplo, para poner los diferentes títulos de sección y otros elementos interactivos que tengan incidencia en el contenido de la pantalla que se está viendo.

Idealmente, como los encabezados están presentes en diferentes secciones de la aplicación, debe haber una consistencia cromática a medida que se navega por las diferentes partes de la app. Sin embargo, otras apps —como la del periódico inglés «The Guardian»— aprovechan los encabezados para jugar con variaciones de color que marcan una diferencia visual entre cada una de las secciones. De esta forma, el encabezado sirve también para extender su identidad visual, y al mismo tiempo darle una pista al usuario de la sección en la que se encuentra.

Colores por defecto de cada sistema operativo

Android y Windows Phone usan temas —*themes* en inglés— que afectan el color de sus aplicaciones y pueden ser claros u oscuros. En Windows Phone la elección entre los temas disponibles está también en manos del usuario, ya que pueden cambiarse en las preferencias del teléfono, modificando la visualización de la interfaz a través de todo el sistema operativo.

Todos los elementos nativos de las aplicaciones se ven afectados por esta decisión: diálogos, botones y fondos de listas se verán en consonancia con el tema elegido. Sin embargo, en Windows Phone, respetar la elección del usuario queda en manos de cada aplicación, porque aun cuando este haya elegido un color para su tema, la app puede decidir usar una opción cromática diferente. Por ejemplo, incluso si el usuario elige un color oscuro como tema, la aplicación puede mostrar elementos en blanco si así se ha establecido previamente.

Mantener o no la selección que hace el usuario del tema, tiene ventajas y desventajas que deben valorarse cuando se decide el diseño de la app. Por ejemplo, usar el color del tema elegido brinda consistencia visual y comodidad a través de todo el sistema operativo, además, el usuario se siente siempre dentro del mismo contexto. No obstante, la diferenciación res-

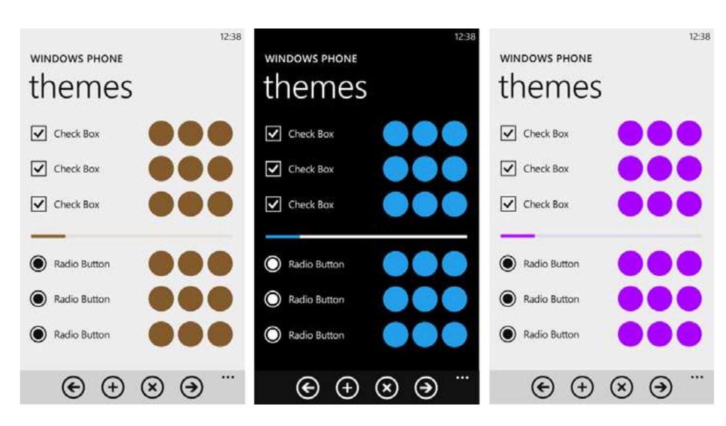


FIG 8.26. Windows Phone deja elegir al usuario temas claros u oscuros y un color de resalte.

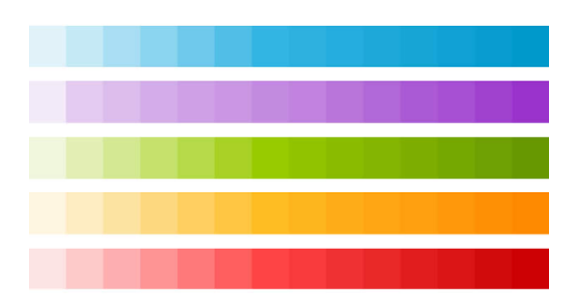
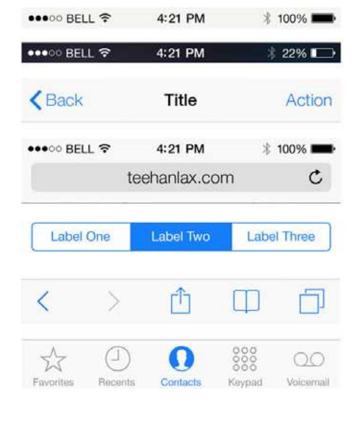


Fig 8.28. *Android sugiere una paleta de colores, aunque su recomendación principal es usar el azul.*

pecto a otras apps puede ser baja y disminuye la posibilidad de incorporar a la aplicación los colores corporativos.

En Android, la elección del tema no es una opción para el usuario; son los diseñadores quienes deben decidir si aplicar temas oscuros o claros a toda la aplicación, o solo a determinadas pantallas.

En los dos casos, los colores de los temas se combinan con otros secundarios que tienen la función de resaltar aquellos elementos que requieren atención. En Windows Phone, este color secundario también puede ser elegido por el usuario dentro de una lista de opciones en las preferencias, pero también en este caso, la app puede haber optado por un color diferente. Mientras tanto, aunque su paleta cromática propone otras opciones que



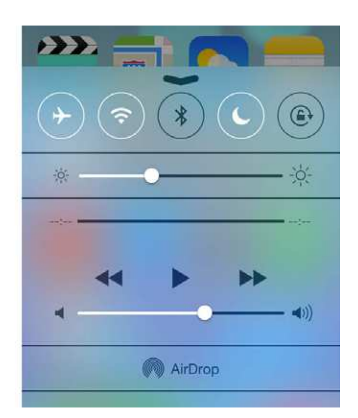


Fig 8.27. *Los colores por defecto de iOS están basados en los negros y azules.*

pueden ser tenidas en cuenta a la hora de plantear el diseño visual, en Android se recomienda el azul como color secundario.

Por último, el caso de iPhone es diferente porque no usa temas. Su propuesta de color nativa, establecida por defecto, se basa en la claridad: blanco y grises muy claros para fondos, barras de herramientas y encabezados de secciones. Estos elementos cuentan también con cierta transparencia que deja ver lo que se encuentran detrás. Mientras que el negro se destina para elementos informativos, el azul se usa como color de resalte para tipografía en etiquetas de botones y formularios, y otros elementos que necesitan destacarse (íconos, pestañas seleccionadas, controles). Aun así, los colores de estos elementos pueden cambiarse fácilmente.

El lenguaje: cuidando las palabras

En una aplicación hay infinidad de lugares donde usar textos: en encabezados y títulos, botones, mensajes de error y avisos en pantallas vacías. En cada caso, las palabras son tan importantes como el elemento gráfico que las contiene o acompaña. Lenguaje textual y visual van de la mano, unidos para lograr una experiencia de usuario consistente en todos los sentidos.

La incidencia de la palabra equivocada

Cómo formular una frase es algo que generalmente no suele considerarse importante en un primer momento, pero tiene influencia directa en la forma en que el usuario usa la aplicación y se relaciona con ella. Por ejemplo, escribir mal la etiqueta de un botón —el texto que contiene— puede llevar a confusiones sobre el resultado de la acción que indica, provocando incertidumbre y frustración en el usuario que se equivoca.

Situaciones como esta son habituales. Aunque pueda parecer obvio, un mensaje debe ser comprendido fácilmente por el usuario, sin obligarlo a esforzarse para interpretarlo. Esto se consigue, entre otras cosas, con un lenguaje simple y directo; haciendo gala de la economía de palabras, donde lo importante se dice primero y lo innecesario, no se dice.

El receptor

Como cualquier mensaje, una aplicación también tiene un receptor. Conocer quién es, permite saber cómo hablarle. Esto es importante para determinar la forma en que se articulan las frases y el lenguaje que se utiliza. No es lo mismo hablarle a un niño que a un adulto, como no es igual dirigirse a un usuario técnico que a un principiante tecnológico. Todo influye en la elección de las palabras.

En el mundo de las aplicaciones se tiende a llevar a un plano muy tecnológico la forma de expresarse, esto hace que muchas veces se escriban mensajes cargados de tecnicismos o palabras complejas que un usuario no preparado podría tener dificultad para asimilar o que suenan demasiado duras. Por ejemplo, la palabra «acceder» puede ser simplemente reemplazada por «entrar», un sinónimo mucho más ameno.

Comunicando errores

La manera de comunicar los errores merece un párrafo aparte. De entrada, esta es una situación estresante o desagradable para el usuario, pero a través de mensajes amables —que no usen tecnicismos incomprensibles o un tono acusatorio— puede sobrellevarse de una forma menos traumática. Un ejemplo de esto, es el sentido del humor para comunicar los errores de



FIG 8.29. Los mensajes de error en Letterpress son tan divertidos como jugar con esta app, quitándole dramatismo al asunto.

Letterpress, que al ser una aplicación de entretenimiento aprovecha la predisposición de sus usuarios a divertirse.

Textos en otros idiomas

Como comentario adicional, se puede mencionar que cuando la aplicación está disponible en diferentes idiomas, hay que tener cuidado al incorporar los textos traducidos para asegurase de que sigan viéndose correctamente en el espacio visual que tienen asignado. Dependiendo del idioma, algunas palabras son —a veces mucho— más largas que en el idioma original y podrían verse cortadas u ocultas bajo otros elementos de la interfaz.

DETALLES VISUALES

Es verdad que la diferencia radica en los detalles. Ellos pueden separar una aplicación regular de una genial. En una primera instancia no se presta atención a estos detalles que mejoran la experiencia del usuario e incluso, cambian su estado de ánimo; y es normal, al principio hay otros aspectos de diseño de los cuales hay que ocuparse. Pero a medida que la interfaz va llegando a su fin, es importante considerarlos para no pasar por alto algo que pueda arruinar el trabajo realizado.

Pantallas vacías

Los diseños tienen que considerar no solo los escenarios ideales, como una lista que ya está completa o una pantalla llena. Es necesario plantear también el diseño para cuando la aplicación comienza a ser utilizada y aún no se dispone de cierta información. Por ejemplo, cómo se vería una pantalla vacía, sin ítems, cuando todavía no hay nada para mostrar.

Otro escenario a tener en cuenta, es aquel relacionado con los fondos que irán detrás de aquellas listas que no alcanzan a cubrir toda la pantalla o los gráficos contenedores donde aparecerán las imágenes una vez que estén cargadas.

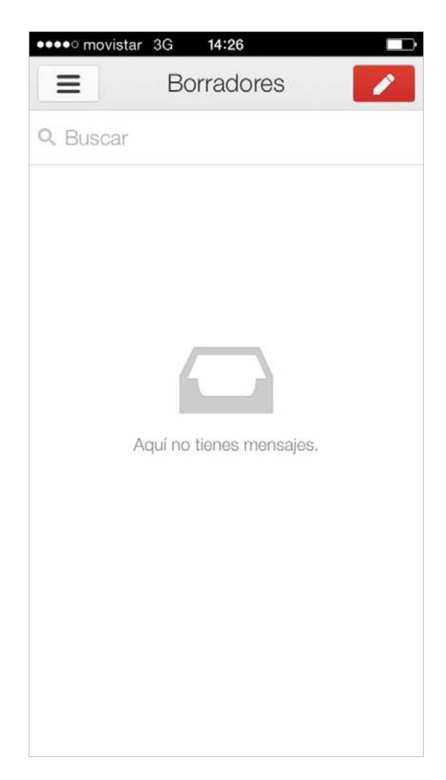


FIG 8.30. También hay que diseñar las pantallas vacías, para cuando aún no hay información.

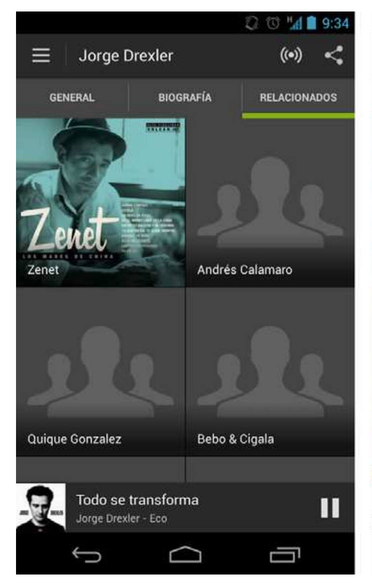




FIG 8.31. De acuerdo a la velocidad de conexión a Internet los elementos pueden tardar en aparecer en pantalla y es bueno considerar qué se verá cuando aún no están visibles.

Gráficos efímeros

Es necesario considerar también el diseño de aquellos elementos que aparecen por poco tiempo en la pantalla. Por ejemplo, si la conexión es lenta, puede ser que el usuario tenga que pasar más tiempo de lo esperado viendo la pantalla de carga y en ese caso, estos detalles visuales tienen su momento de gloria.

Secretos visuales

A la hora de darse pequeños lujos visuales, se puede pensar en aquellos gráficos que están ocultos a primera vista, que no son de fácil acceso para el usuario, pero que una vez descubiertos generan un placer especial ayudados por la complicidad. Por ejemplo, en la aplicación de Google+[9] cuando se arrastra y suelta

9 http://www.google.com/mobile/+/

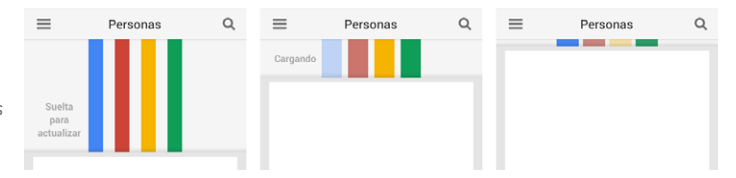


Fig 8.33. Google+ aprovecha espacios ocultos para añadir un toque de color.

la pantalla para actualizar los contenidos, aparece una serie de tiras de colores, un detalle no visible hasta que realizamos esta acción.

Animando la app

Las animaciones son más difíciles de diseñar y, en ocasiones, de percibir que los llamados detalles visuales, pero también hacen una diferencia en la experiencia de usuario[10], pues llenan de vida la aplicación y hacen más agradable su uso, a la vez que acompañan una función principal o una acción. Seguramente no son fundamentales, porque lo mismo podría representarse sin estar animado, pero sin duda influyen mucho en cómo se siente el usuario al interactuar con los elementos en pantalla.

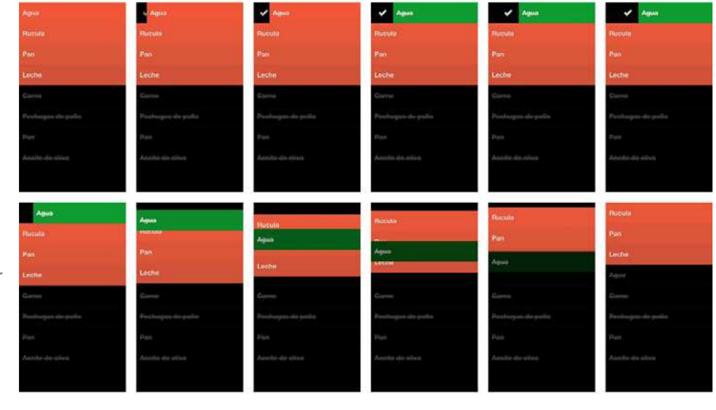


Fig 8.32. Clear usa las animaciones como indicador para saber qué pasa con las tareas una vez que se completan.

10 Jonhson, Ben. Great animations make great apps. http://www.raizlabs.com/2012/09/great-animations-great-apps/

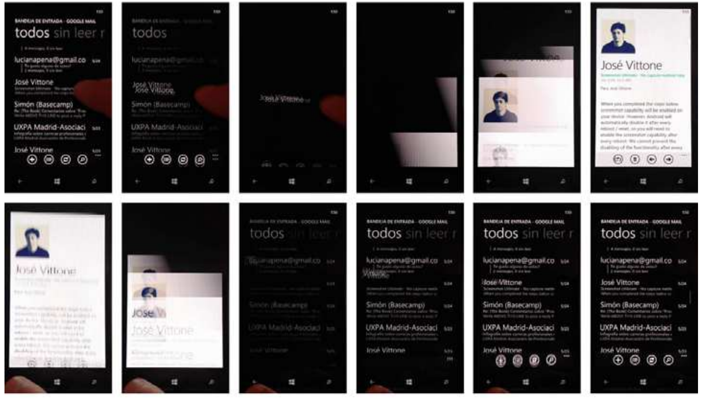


FIG 8.34. En Windows Phone abundan las transiciones entre pantallas, enriqueciendo la experiencia de navegación.

De acuerdo a su función o a la forma de utilizarlas, las animaciones pueden servir para diferentes cosas:

1. Como *feedback*. Los gráficos en pantalla pueden animarse para mostrar el resultado de una acción. Tal sería el caso de un elemento que desaparece de la pantalla después de ejecutarse una acción, mostrando hacia dónde va y qué ha pasado con él.
2. Como transiciones. Las animaciones pueden ayudar a hacer las transiciones entre pantallas más fluidas. Esto puede representar una buena experiencia de navegación cuando se va de un contenido hacia otro.
3. Como herramientas informativas. Especialmente cuando se usa la app por primera vez, puede ser necesario mostrar algún elemento en la pantalla señalando cómo se usa o destacar la presencia de un componente nuevo que no existía en versiones anteriores. En casos como estos, la animación puede ayudar para dirigir la atención, que es, de hecho, una de sus funciones principales.
4. Puro «caramelo visual». A veces, una animación no cumple otra función más que «endulzar» el ojo del usuario. Puede ser un simple detalle que no todos vean, pero que sin duda causará una buena impresión visual al ser notado.



9

Dustin Mierau: Path y el valor de los detalles

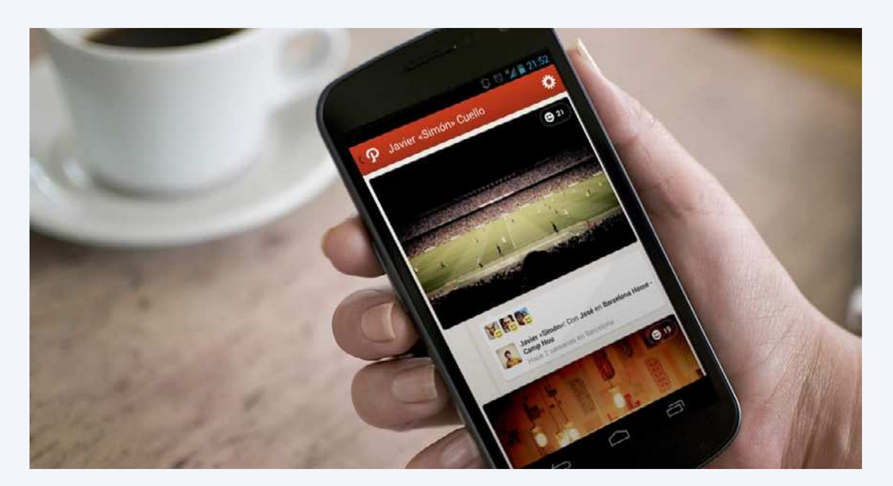


FIG 9.1. *El tratamiento visual de la interfaz de Path cuida al milímetro cada elemento que la compone.*

Path es una red social focalizada en móviles que, a diferencia de otras, permite agregar un máximo de 150 personas a tu grupo de amigos para compartir «momentos», como si fuera un diario personal. Aunque no es tan conocida en algunos países, ya ha superado la barrera de los veinte millones de usuarios.

Hemos usado Path desde sus primeras versiones y, desde entonces, nos ha encantado el tratamiento visual y la importancia que le dan a los pequeños detalles en la interfaz. El reloj moviendo sus agujas cuando te desplazas por el listado de eventos en la pantalla principal, es un ejemplo de estos detalles que arrancan una sonrisa y te hacen sentir bien al utilizar la app, mejorando notablemente la experiencia de uso.

Entrevistamos a Dustin Mierau, cofundador y Jefe de Diseño de Path, para que nos explicara un poco mejor la importancia y el papel de estos elementos.

¿Qué peso crees que tienen estos toques en el diseño general de la interfaz?

> *Definitivamente requiere un balance. Nos divertimos mucho añadiendo caprichos a nuestros productos, pero somos cuidadosos en asegurar que el producto sea primero,*

útil; después, intuitivo y, finalmente, convincente. El reloj nos ayudó a quitar el desorden del listado de eventos, permitiéndonos mostrar más historias en la pantalla al mismo tiempo: útil. Menos información en modo texto —como las fechas y horas— hizo que lo que está en pantalla se lea más intuitivamente. Y, con un reloj analógico, pudimos agregar la sensación de movimiento a través del tiempo, hacia atrás y hacia adelante, haciendo el listado de eventos de Path más convincente.

Este tipo de ideas pueden parecer conceptos difíciles de transmitir al resto del equipo, especialmente en las primeras versiones de la app que están más enfocadas en lograr una buena funcionalidad y donde estos elementos visuales suelen dejarse «para más adelante».

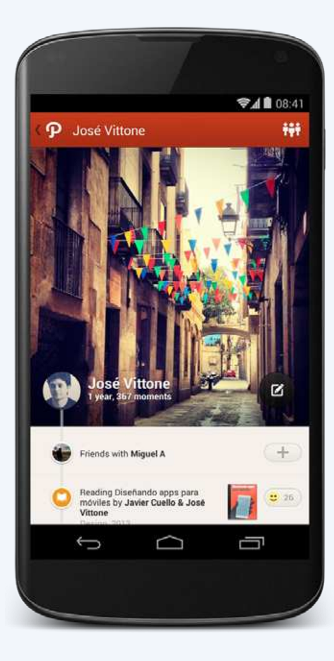
¿Cómo crees que debería ser la interacción entre diseñadores y programadores? ¿Encuentras difícil comunicar tus ideas a los desarrolladores?

No es difícil en absoluto. En Path hay confianza entre los equipos. Las ideas más elaboradas necesitan algo de convencimiento —y con razón— pues usualmente requieren mucho más tiempo y esfuerzo para implementarse. Nuestros ingenieros y diseñadores están aquí para desarrollar grandes productos y cómo funciona un producto es el resultado de una gran colaboración y comunicación entre ellos. Mientras más respeto haya entre estos equipos, mejor funcionará el producto. Normalmente, se puede percibir la fricción de un equipo a través de su producto.

Tener que decidir entre una interfaz bien terminada o desarrollar una base sólida que garantice el correcto funcionamiento de la app a veces parece una decisión de «una u otra, pero no las dos».

Ahora que Path se vuelve cada vez más compleja, ¿es difícil mantener una buena funcionalidad sin sacrificar la experiencia de usuario o viceversa?

> *Es verdad que Path tiene muchas más funciones hoy que hace dos años, pero hemos considerado cada función que añadimos a nuestros productos. Hay muchas cosas que no hemos agregado porque no hemos encontrado una buena forma de incorporarlas: tener un montón de funciones no significa mucho si no puedes descubrir cómo usarlas.*
>
> *Nos esforzamos por encontrar la manera correcta de añadir funciones al producto. Se agregan cuando actúan bien por sí solas y combinadas con otras funciones de la app, creando una experiencia convincente, útil e intuitiva.*



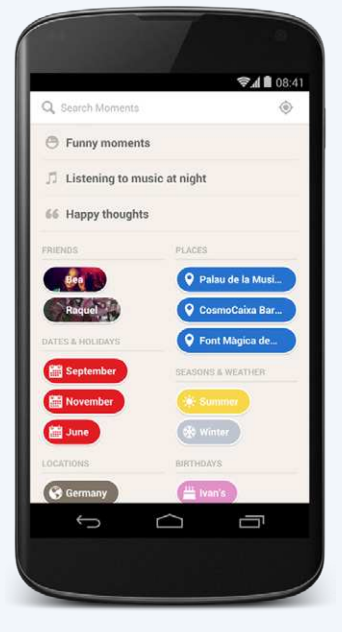


FIG 9.2. *En Path, cada función que se añade a la app tiene su razón de ser y se evalúa con detenimiento.*

Luego de las aplicaciones para iPhone y Android, Path lanzó una versión para iPad que, comparada con las otras, es relativamente nueva y aún está en camino de definir su personalidad.

La app de Path para iPad ha reutilizado algunos patrones de diseño que ya estaban presentes en iPhone, pero otros son nuevos. ¿Qué crees que es lo mejor de tener un montón más de espacio para jugar?

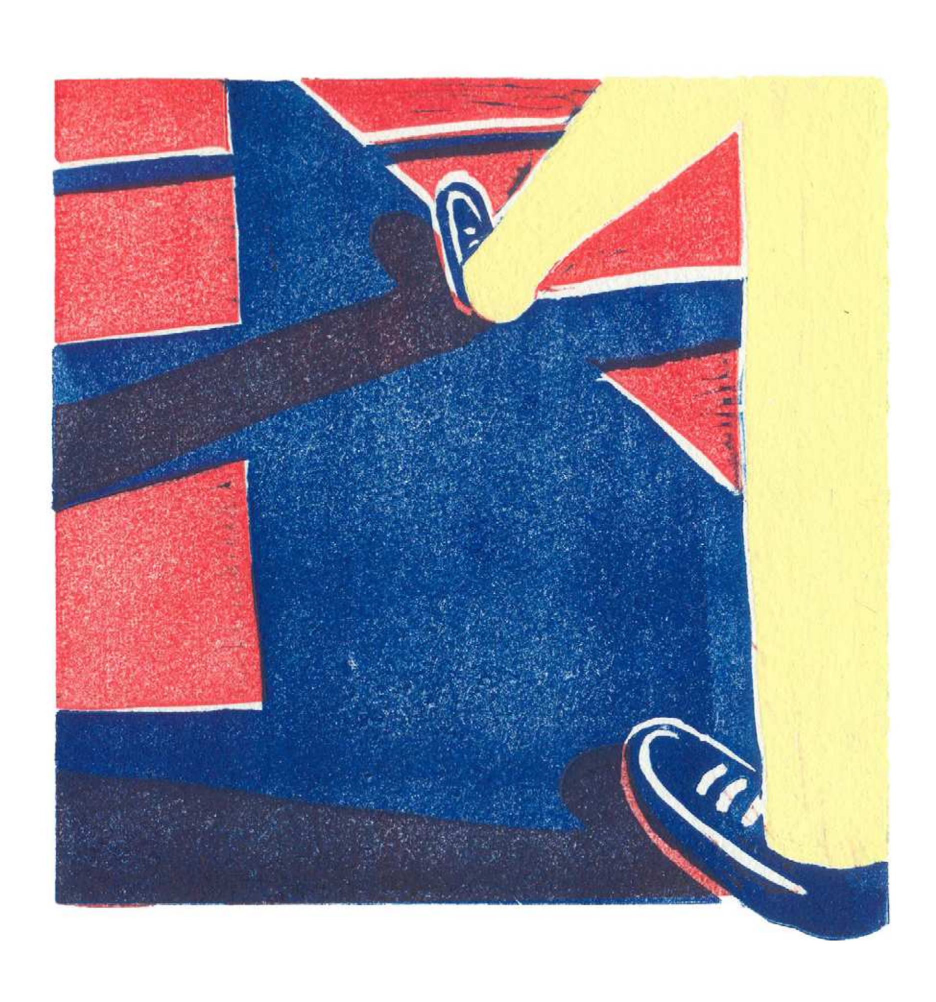
> *No es solo que tenemos más espacio para jugar, es un caso de uso completamente diferente. Cuando usas Path en tu iPhone o iPod Touch, posiblemente estás ocupado, en el trabajo, en un viaje, afuera, con amigos, etc. Cuando estás usando el iPad, estás en casa, en un café, en el trabajo y probablemente sentado. Estás en un mejor lugar para consumir más, posiblemente con tiempo para interactuar más y el tipo de contenido que estás listo para crear es único. Con el iPad estás preparado para compartir una colección de fotos, escribir pensamientos más profundos, tomarte el tiempo para editar vídeos, simplemente, para ser más creativo.*
>
> *Apenas hemos empezando a explotar lo que Path puede hacer en iPad. Creo que el equipo espera aprovechar el espacio que el dispositivo ofrece para permitir a nuestros usuarios ser mucho más creativos en la forma de compartir sus vidas.*

Sin dudas, esta aplicación es una buena referencia para diseñadores y desarrolladores, que pueden encontrar en ella un ejemplo de cómo lograr un producto de calidad sin descuidar ningún aspecto. Para muchos diseñadores puede parecer difícil de alcanzar, sobre todo cuando recién están comenzando los proyectos y el camino hacia adelante no está del todo claro.

Finalmente, Dustin aconseja a todos aquellos que se están iniciando en el mundo de las apps y pueden estar algo perdidos por la infinidad de aplicaciones y sistemas operativos diferentes:

> *Encuentra una utilidad, deshazte de todo lo que no tenga sentido y hazla tuya. Absorbe tanto conocimiento técnico como puedas. Sé humilde. Sé curioso. Respeta a los ingenieros. Respeta a tus usuarios.*

Visita la web de Path para conocer más de la aplicación y por qué no, para descargarla y entender por qué a nosotros nos encantó: **www.path.com**



10

Probando con usuarios

Los test de usabilidad son una herramienta fundamental para corregir y mejorar la aplicación. Se llevan a cabo con base en la observación de los usuarios: cómo interactúan con ella y qué tan fácil les resulta usarla.

Qué son los test de usabilidad

A medida que se diseña la aplicación, el equipo necesita comprobar si a los usuarios les resulta fácil de usar y si cumple la finalidad para la que fue pensada originalmente; por eso, hace falta realizar pruebas con ellos.

Estos test de usabilidad se llevan a cabo para obtener retroalimentación de los usuarios, de forma tal que, tras la observación de su comportamiento al usar la app, sea posible corregir y mejorar aspectos de usabilidad[1]. Los test se efectúan en laboratorios o espacios preparados para tal fin, donde un usuario voluntario realiza tareas concretas, previamente establecidas por un moderador que guía la prueba, mientras un grupo de observadores realiza anotaciones acerca de lo que ve.

Qué no son estos test

Los test de usabilidad no deben confundirse con los llamados focus group, que tienen un objetivo completamente diferente, ya que, en estos últimos, un grupo de personas expresa sus opiniones acerca de los diseños que se le muestran, sin tener en cuenta si la app sirve o no para realizar una tarea específica.

En qué etapa del proyecto deben hacerse

Es ideal que los test se realicen antes de lanzar la app, incluso mejor si es antes de pasar a la programación del código. Probar en etapas tempranas y de forma asidua, puede ayudar a obtener ideas antes de la publicación, cuando aún se está a tiempo de implementarlas, lo cual tiene la ventaja de ahorrar tiempo y dinero al hacer cambios que puedan tener impacto en el proyecto.

1 ARMENGOL, Daniel. ¿Qué es un test de usabilidad? http://www.usolab.com/wl/2012/08/que-es-un-test-de-usabilidad.php

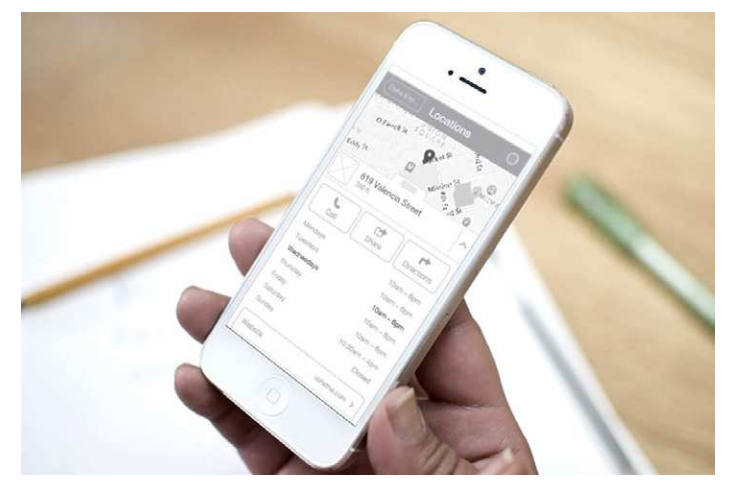


FIG 10.1. No es necesario tener una app terminada para hacer pruebas con usuarios.

Aunque generalmente se piensa que esto se hace una sola vez, los test pueden llevarse adelante las veces que sea necesario, incluso, cuando la app ya ha sido lanzada, para seguir obteniendo información de los usuarios que ayude a mejorarla.

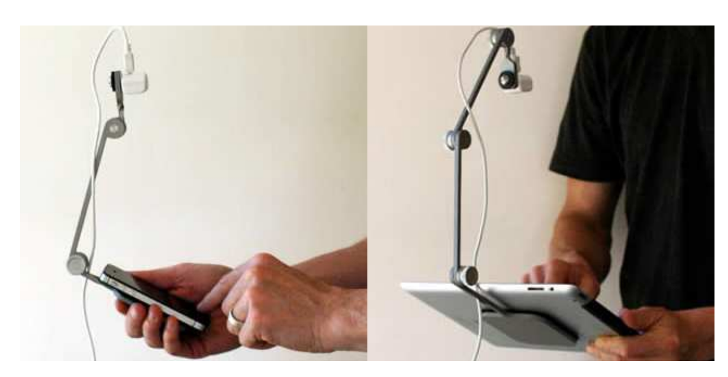
TEST EN MÓVILES

Cuando se realiza un test de usabilidad en un móvil, los usuarios no necesariamente tienen que probar una aplicación completamente terminada. De hecho, como hemos comentado anteriormente, lo más aconsejable es realizar estos test en etapas tempranas del proceso de diseño y desarrollo, por ello, un prototipo puede ser suficiente para las pruebas[2].

Las aplicaciones para móviles están vinculadas a un contexto de uso determinado, algo difícil de replicar a menos que se realice un test de campo. Hay que tener en cuenta que probar la aplicación en una habitación puede no ser el escenario más

2 CASSANO, Jay. Secrets From Facebook's Mobile UX Testing Team. http://www.fastcolabs.com/3007979/open-company/secrets-facebooks-mobile-ux-testing-team

FIG 10.2. Es importante también grabar los gestos en las pruebas. Mr. Tappy ayuda a conseguirlo.



realista, pero claramente es mejor que no realizar ningún test en absoluto. En este caso, hay que intentar simular las condiciones de uso lo más cercano a la realidad posible, por ejemplo, hacer la prueba con una conexión a Internet promedio —como la que tiene la mayoría de la gente— en lugar de una Wi-Fi que funcione a gran velocidad[3].

El teléfono también debe acercarse a la realidad, lo ideal entonces sería realizar la prueba en el móvil del usuario. Si no es posible, hay que conseguir el terminal que más se le parezca y explicarle previamente su funcionamiento para evitar que las características propias del teléfono afecten el desarrollo del test, permitiendo extraer conclusiones específicas del uso de la app. De manera similar, si la aplicación que se prueba es, por ejemplo, para Android, el usuario también debe ser usuario habitual de este sistema operativo.

Por último, en el caso de los móviles es necesario considerar algo que para la web no se tenía en cuenta: los gestos de los dedos. Para esto se pueden instalar dispositivos de filmación en el teléfono que, además de grabar dónde y qué toca el usuario, también pueden registrar cuáles son los gestos que realiza para completar las acciones[4].

3 LANG, Tania. Eight Lessons in Mobile Usability Testing. http://uxmag.com/articles/eight-lessons-in-mobile-usability-testing

4 http://www.mrtappy.com/

Test guerrilla

Los test guerrilla son una alternativa ágil y económica a los test de usabilidad tradicionales que requieren, entre otras cosas, profesionales especializados, gran cantidad de usuarios y espacios de trabajo que incrementan los costes. Este tipo de test, con un formato más informal y menos usuarios, puede ser también eficaz y es especialmente útil cuando la fecha límite para terminar el proyecto está cerca[5].

Un test guerrilla consiste en reunir a una determinada cantidad de usuarios para probar la aplicación, con el objetivo principal de ver cómo se comportan y obtener información que permita corregir errores[6].

Cómo hacer un test guerrilla

Para llevar a cabo el test, hace falta designar a una persona del equipo que será responsable de liderar la prueba. Para este rol, la paciencia y capacidad de escuchar atentamente a los usuarios son fundamentales para guiarlos durante el proceso.

No obstante, es ideal que junto con el responsable participen más miembros del equipo —al menos otra persona— como observadores. Su trabajo se limitará a tomar nota de los problemas de usabilidad que puedan aparecer, sin tener una participación activa.

El desarrollo completo del test se divide en tres etapas bien diferenciadas: preparación, ejecución y análisis.

5 http://stamfordinteractive.com.au/conversations/benefits-of-guerrilla-testing/

6 Chisnell, Dana. Usability Testing Demystified. http://alistapart.com/article/usability-testing-demystified

Preparación

En esta etapa se sientan las bases del test antes de la reunión con los usuarios. Aquí se establece el objetivo —como probar toda la app o solo algunas partes— y se definen las pautas necesarias para sacar el máximo provecho.

También en esta etapa del proceso se elige a los participantes. Lo ideal es contar con un número de voluntarios que oscile entre cinco y ocho, pues con esta cantidad de personas se puede detectar casi la totalidad de problemas de usabilidad más comunes. Incrementar el número de participantes no garantiza que la cantidad de problemas encontrada sea mayor.

Además de los usuarios de prueba, también hace falta definir el lugar donde se llevará a cabo el test. Preferiblemente, tiene que ser un sitio tranquilo, donde no haya interrupciones ni distracciones externas.

Durante la fase de preparación, hay que decidir cuál de las dos pruebas posibles se hará a los voluntarios. La primera alternativa consiste simplemente en mostrar a los usuarios la app y ver si entienden para qué sirve y cómo funciona. En el segundo tipo de prueba, se les asigna una tarea específica y se observa cómo la resuelven con la aplicación.

Con base en la decisión anterior, pueden definirse las preguntas que se harán para obtener información acerca de sus acciones. Estas preguntas deben ser neutrales y de acuerdo al



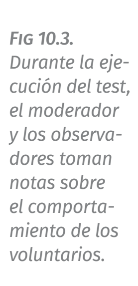


FIG 10.3. *Durante la ejecución del test, el moderador y los observadores toman notas sobre el comportamiento de los voluntarios.*

conocimiento de los usuarios. Una recomendación es romper el hielo con preguntas fáciles de contestar, para evitar intimidar a los participantes desde el primer momento.

Una vez que se tenga lo anterior definido, se puede hacer una pequeña prueba piloto entre los miembros del equipo para completar la preparación del test, antes de realizar la prueba final con usuarios.

Ejecución

El test de usabilidad se realiza con un voluntario a la vez. Antes de comenzar la prueba es necesario hacer una pequeña introducción, para explicarle en qué consistirá el test y cuánto tiempo durará. Además, durante esta explicación, hay que invitar al voluntario a participar activamente durante el proceso, motivarlo a decir lo que está pensando mientras realiza las acciones y a expresar sus preocupaciones y problemas, aun cuando crea que se trata de críticas. Es importante que el participante sienta que sus opiniones son valiosas, de gran utilidad para el proyecto y que de ninguna forma afectarán los sentimientos del equipo.

Durante el desarrollo del test, el moderador tiene que estar más atento al comportamiento del usuario que a lo que dice. Esto debe ser así, porque es normal que a veces los participantes mientan sin intención o intenten complacer al responsable para quedar bien.

Cuando la prueba consiste en realizar una serie de tareas con la aplicación, el moderador debe proponer al usuario llevarlas a cabo sin asistencia. Es importante no sesgar ni influir en su comportamiento. Solo en situaciones que lo requieran, es posible una pequeña orientación para ayudarlo a seguir; en ese caso, se le puede preguntar qué piensa o cuál es el resultado esperado de alguna acción que acaba o está a punto de ejecutar.

El moderador puede no ser la única persona que haga preguntas durante el test: es posible que el participante tenga dudas y es necesario hacerle notar que se pueden responder sus inquietudes una vez acabada la prueba.

Al concluir esta etapa, es recomendable dar a cada participante una pequeña recompensa por haber formado parte de la prueba. Si bien puede mencionarse con antelación, se sugiere dar esta retribución al final para no condicionar las respuestas de los usuarios.

Análisis

Una vez que las pruebas han finalizado, sin dejar pasar demasiado tiempo, es importante que moderador y observadores se reúnan para comentar los resultados del test y revisar las notas obtenidas durante las sesiones con usuarios.

En esta etapa se seleccionan los problemas que serán resueltos e inmediatamente se comienzan a plantear las posibles soluciones. Cuando los problemas se hayan corregido, puede realizarse otra prueba para verificar que las soluciones propuestas sean realmente efectivas o determinar si han aparecido nuevas situaciones conflictivas.

Otras formas de obtener información

Los test tipo guerrilla no son la única manera de obtener información de los usuarios. Hay caminos mucho más rápidos e incluso, algunos no requieren más que unos pocos segundos.

Dogfooding: Comiendo tu propia comida

Una leyenda urbana cuenta que, a raíz de un comercial de televisión de comida para perros, un ejecutivo de Microsoft envió un correo a sus empleados diciendo que debían probar su propia comida, o mejor dicho, usar sus propios productos[7].

7 http://www.elqudsi.com/articulos/%C2%BFque-es-el-dogfooding/

Desde entonces, el *dogfooding* se ha convertido en una forma fácil y rápida de probar el software de producción propia, ya que es tan simple como usar la aplicación en la que se ha estado trabajando para compenetrarse con ella.

Con el tiempo, puede transformarse en una práctica habitual para detectar errores técnicos y conceptuales que pueden resolverse inmediatamente, al tener un conocimiento casi perfecto sobre ellos. Este tipo de problemas suele ser difícil de detectar cuando se está diseñando o programando el código, pero sí es posible que algunos detalles salten a la vista cuando se está abstraído y se prueba la aplicación en un entorno más relajado.

Aunque es una práctica tan simple que puede llevarse a cabo en el lugar de trabajo y con los compañeros de equipo, usar la app que se está diseñando es una herramienta que no reemplaza el test con usuarios, pero sí sirve para probarla y demostrar confianza en lo que uno mismo hace y produce.

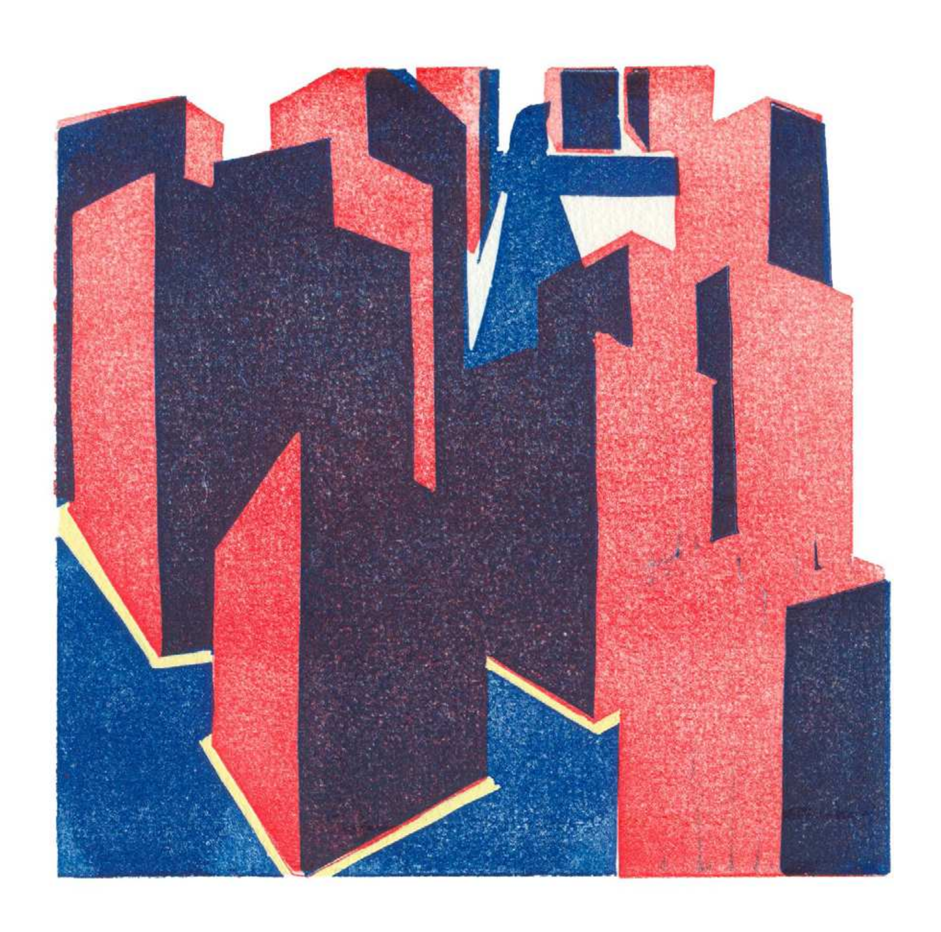
Test de los cinco segundos

La primera impresión sobre un diseño aporta mucha información valiosa, principalmente sobre jerarquías de contenidos. El test de los cinco segundos consiste en explicar al usuario una situación concreta e inmediatamente después, enseñar el diseño de la aplicación por solo cinco segundos. A continuación, se pide al participante que diga rápidamente todo lo que recuerda de lo que ha visto.

Las reacciones del usuario luego de este rápido vistazo ayudan a establecer qué cosas le resultan más llamativas para determinar si coinciden con lo que el equipo había pensado inicialmente.

Existen herramientas en Internet para llevar a cabo esta prueba como la web FiveSecondTest[8].

8 http://www.fivesecondtest.com/



11

Preparando los archivos

Una vez definido el diseño de la interfaz, la siguiente etapa –quizás algo más tediosa– consiste en separar y preparar los archivos para que queden listos para el desarrollador. Hacer esto de la forma correcta favorecerá una implementación fiel del diseño.

ENTENDER LAS DIFERENCIAS TECNOLÓGICAS

En el mercado hay infinidad de móviles con características de hardware que varían dependiendo del fabricante. De esas posibles variaciones, las que más afectan el trabajo del diseñador son aquellas relacionadas con la calidad de la pantalla, ya que influirán en la forma de plantear el diseño de la interfaz y más tarde, en las imágenes que se prepararán para el desarrollador.

Tamaño de pantalla

Un factor determinante en las pantallas de los dispositivos móviles es su tamaño: la distancia, medida en pulgadas, de un extremo a otro en forma diagonal.

En este sentido, Android se trata de un sistema operativo muy fragmentado debido a la gran cantidad de modelos que ofrecen los fabricantes, con pantallas que varían de un teléfono a otro sin mayor coherencia; esta es una de las consecuencias de ser un sistema operativo abierto y disponible en muchos terminales. Ahora bien, para poner un poco de orden en este caos, Android ha decidido agrupar los distintos tamaños de

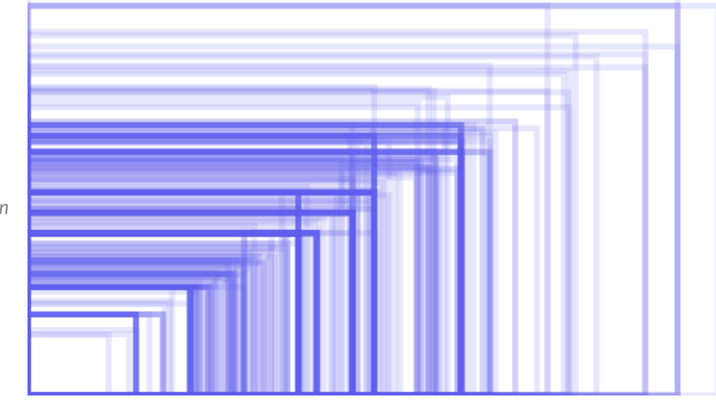


FIG 11.1. En Android existe una inmensa fragmentación debido a los numerosos tamaños de pantalla de cada fabricante en sus teléfonos.

pantalla en cuatro categorías principales: pequeña, normal, grande y extra grande.

En el mundo iOS las cosas están más o menos en orden: iPads, iPads mini y iPhones, tienen especificaciones de pantalla sin sobresaltos, ya que todos se encuentran bajo el control de Apple, único fabricante de los terminales que llevan este sistema operativo.

Hasta la fecha no existen muchos fabricantes que provean Windows Phone y los tamaños de pantalla se limitan a tres tipos, cambiando ligeramente la relación de aspecto entre ellos: WVGA (15:9), WXGA (15:9) y 720p (16:9).

Densidad de pantalla

Además del tamaño, otro factor a tener en cuenta es la densidad de la pantalla. Esta se refiere a la cantidad de píxeles que entran en un determinado espacio físico y se mide en puntos por pulgada —DPI por sus siglas en inglés—, donde un punto equivale a un píxel. Para entender un poco mejor, la densidad de pantalla puede compararse con la densidad de población de un país —aquella que relaciona la cantidad de habitantes por kilómetro cuadrado—; en el caso de las apps, los habitantes serían los píxeles y la superficie, la pantalla. Una mayor densidad de pantalla indica mayor cantidad de píxeles disponibles y, por lo tanto, mejor calidad de definición de imágenes y otros elementos que componen la interfaz.

La densidad influye en el trabajo del diseñador porque determina las características del documento con el que se empezará a diseñar y la cantidad de imágenes que se deberán producir al terminar el diseño: a mayor cantidad de densidades soportadas por un sistema operativo, mayor será el número de imágenes que harán falta.

En Android hay cantidad de densidades que cubren los móviles y tabletas, ofreciendo un amplio abanico de calidades de

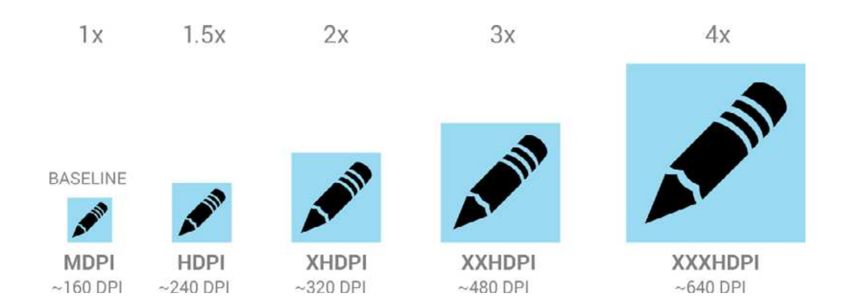


FIG 11.2. Es necesario diseñar una imagen para cada densidad en la que funcionará la aplicación.

visualización. Se encuentran agrupadas en baja, media, alta o extra alta[1].

iOS solo cuenta con dos densidades: retina y no retina. La primera es más actual y ha comenzado a utilizarse para casi todos los dispositivos recientes, a excepción de iPad mini. Sus características visuales duplican la cantidad de píxeles —o puntos— por pulgada que se encuentran en los dispositivos que no cuentan con esta tecnología.

Por último, Windows Phone separa las densidades en media, alta y HD, dependiendo del terminal.

Preparando el documento para comenzar a diseñar

Cuando se diseña una interfaz, claramente el primer paso es enfrentarse a un documento en blanco y aquí asalta la primera duda. ¿Qué tamaño debería tener el espacio de trabajo?

Android recomienda empezar a diseñar con un documento preparado para una base estándar —que surge de combinar un tamaño de pantalla normal con una densidad media— y después, a la hora de separar las imágenes para el desarrollador, escalarlas para densidades mayores o menores.

1 http://developer.android.com/guide/practices/screens_support.html

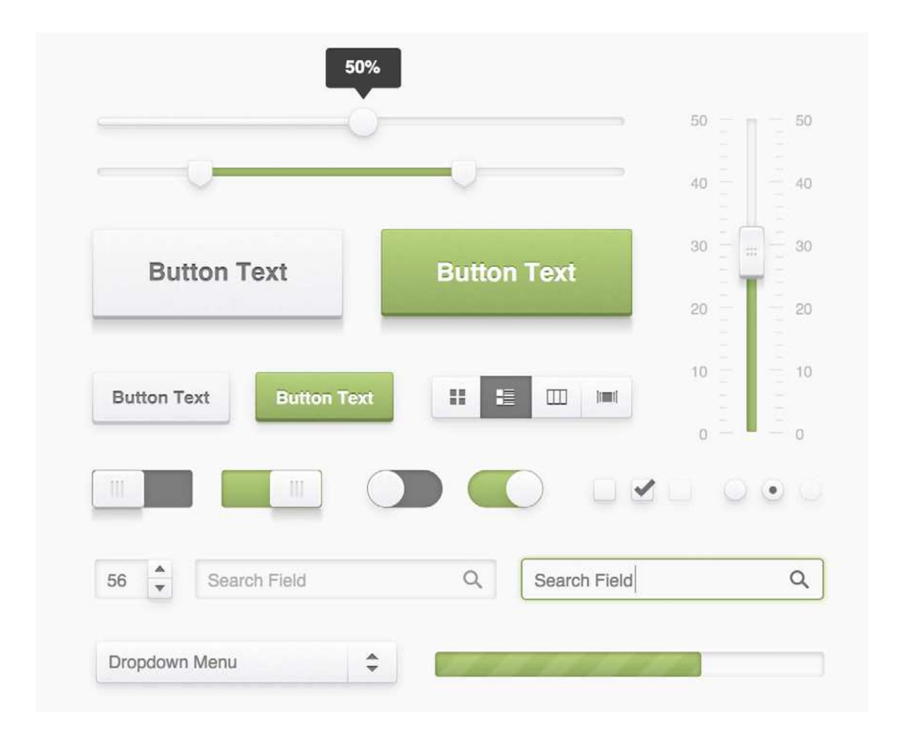


FIG 11.3. Existen plantillas con componentes de interfaz que se pueden utilizar de base.

Por el contrario, en iOS lo más conveniente es diseñar en retina, la densidad mayor, y en Windows Phone, hacerlo a partir de la menor de las densidades disponibles, WVGA, y de allí escalar a las dos restantes.

Por suerte, no siempre hay que diseñar desde cero. Existen en Internet plantillas de diseño tanto para Android[2], como para iOS[3] y Windows Phone[4], que contienen elementos de interfaz que pueden ser aprovechados para comenzar con una base de componentes visuales como botones, barras y fondos. La mayoría de estos gráficos están en formato vectorial, lo que facilita llevarlos a diferentes tamaños.

2 http://developer.android.com/design/downloads/index.html
3 http://www.teehanlax.com/tools/ios7/
4 http://msdn.microsoft.com/library/windowsphone/design/ff637515(v=vs.105).aspx

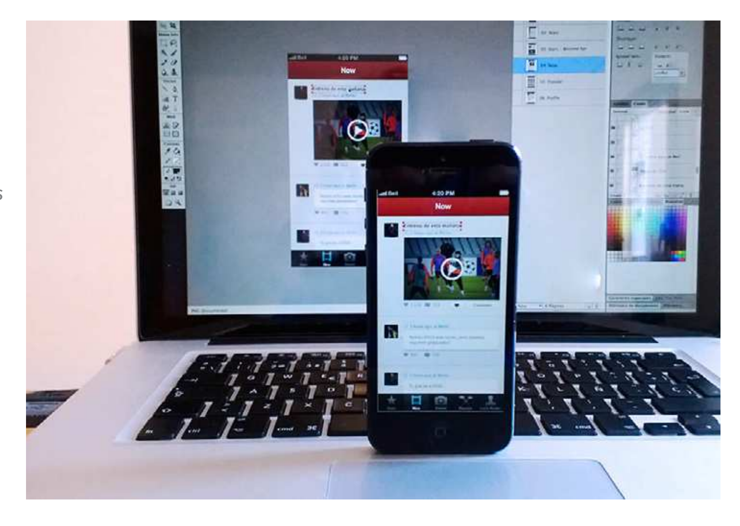


Fig 11.4. *LiveView permite comprobar cómo se ve en el teléfono el diseño mientras se trabaja.*

Diseñar y probar en el móvil

Una interfaz puede verse muy bien en la pantalla del ordenador, pero esto no es un escenario real pues no es allí donde finalmente se verá el diseño. Es conveniente, a medida que se va trabajando, probar constantemente en un teléfono para verificar la visualización y no esperar a tener implementada la aplicación, después del código, para llevarse sorpresas.

Acostumbrarse a hacerlo e incorporarlo de forma habitual en el proceso de diseño permitirá, entre otras cosas, asegurarse de que el tamaño de los elementos y la tipografía son correctos y que los contrastes son suficientes.

Existen diferentes herramientas que permiten probar el diseño directamente desde el ordenador conectado al móvil. Para Android una alternativa es usar Android Design Preview[5], mientras que en iOS su equivalente es LiveView[6]. Por otro lado, en Windows Phone no existe ninguna herramienta similar.

5 https://code.google.com/p/android-ui-utils/
6 http://www.zambetti.com/projects/liveview/

Separar las imágenes

Una vez que el diseño está listo es necesario separar las imágenes incluidas en la interfaz, guardando cada elemento en un archivo separado, para que el desarrollador pueda utilizarlas y comenzar a darle vida a la aplicación.

Una imagen para cada densidad

Cuando se separan las imágenes hay que considerar las diferentes densidades disponibles y preparar una versión para cada una de ellas.

Si se cubren todas las densidades, en Android deberían obtenerse cuatro imágenes distintas por cada elemento gráfico. Si se ha diseñado en una base estándar, como comentamos antes, hará falta hacer un pequeño cálculo, multiplicando anchos y altos de las imágenes, para que se vean bien en otras densidades.

Para diseñar y no perderse entre las posibilidades, Android trabaja con «píxeles independientes de densidad», mejor conocidos como DP. Cada DP equivale a `1px` en una pantalla de densidad media y puede trasladarse a otras densidades calculando su valor para cada caso. Por ejemplo, si tenemos un ícono de `20dp` de ancho y alto, este tendrá en una densidad media, `20px`; en una densidad baja, `15px` (`20dp` multiplicados por 0.75); en una densidad alta, `30px` (`20dp` multiplicados por 1.5) y finalmente, en una densidad extra alta, `40px` (`20dp` multiplicados por 2).

Afortunadamente, no hace falta perder la cabeza con una calculadora ni ser científico nuclear ya que existen herramientas en Internet para hacer este trabajo[7].

La cuenta en iOS es mucho más fácil: simplemente hay que llevar las imágenes diseñadas en retina a la mitad de su

7 http://coh.io/adpi

tamaño. Como secreto del oficio, es importante considerar que las medidas de ancho y alto de los gráficos diseñados en retina sean números pares, para que después puedan ser fácilmente divididas por dos.

En Windows Phone las imágenes se trabajan como vectores, con el formato `.svg` y de esta forma, se pueden escalar por código al tamaño necesario. Esto también garantiza el aprovechamiento de las mismas en caso de querer hacer una app para Windows 8, asegurando la consistencia visual entre las diferentes aplicaciones.

Al escalar las imágenes es necesario revisar que estas mantengan su calidad y que no se deformen o se vean borrosas. Para conseguirlo, a veces hace falta ajustar las imperfecciones en el nuevo tamaño, hasta lograr una imagen nítida y bien definida, una técnica también conocida como pixel perfect.

Botones

Para los botones no es conveniente trabajar con imágenes que contengan la forma completa del botón en su tamaño final; más bien, tienen que estar preparadas para poderse estirar y cubrir diferentes anchos y altos sin deformarse. De esta forma, puede usarse la misma imagen para botones de diferentes tamaños a través de toda la aplicación, evitando tener varias imágenes que haya que volver a diseñar y exportar ante cualquier cambio. ¡Vaya fastidio!

Para facilitar este trabajo y decidir el control sobre las áreas de la imagen que se quieran estirar, en Android existe una herramienta llamada Draw 9-patch[8] que sirve para estos menesteres. El resultado final es una imagen que, luego de ser implementada, detectará cuáles zonas puede estirar y cuáles debe conservar intactas.

8 http://developer.android.com/tools/help/draw9patch.html

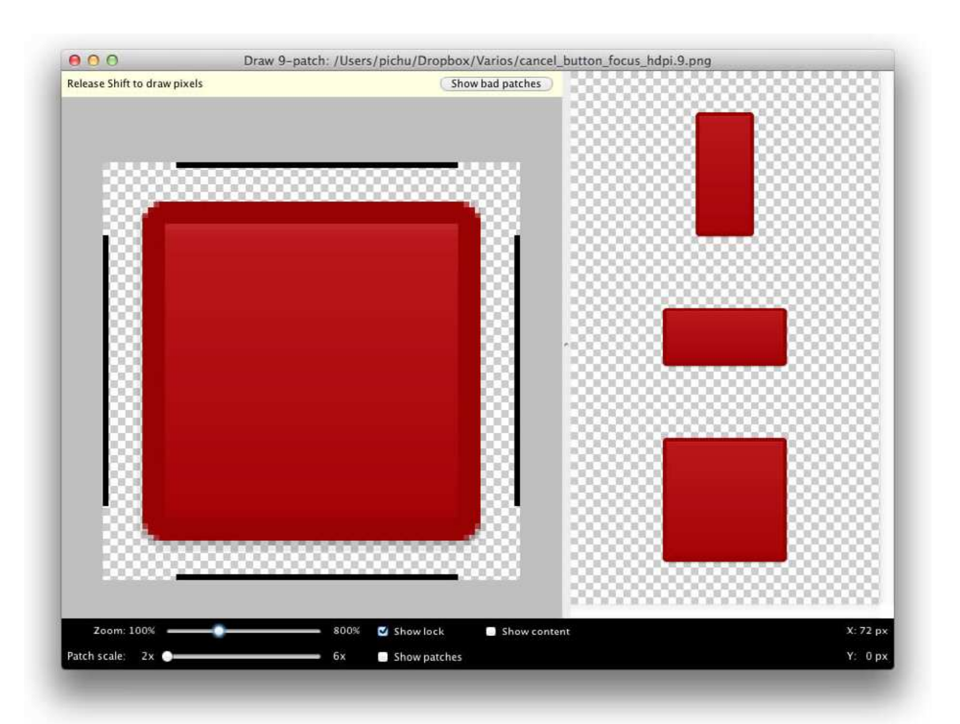


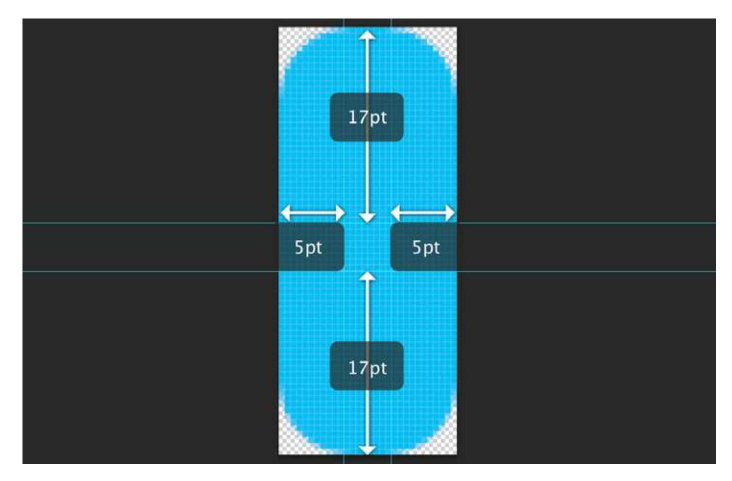
FIG 11.5. *Draw 9-Patch permite determinar cómo se escalará la imagen al estirarse.*

En iOS la idea es parecida, solo que no hace falta una herramienta externa y puede contemplarse desde que se diseña la imagen. Por ejemplo, en un botón con esquinas redondeadas, puede definirse un área de seguridad que no se estire y usarse el centro del botón para cubrir el alto y ancho necesarios. Estos valores deben acordarse con el desarrollador porque deben ser ingresados por código en el momento de programar la aplicación[9].

Es necesario recordar que los botones no tienen solamente un estado. Por lo general se diseña el estado «normal», pero también pueden estar presionados o deshabilitados —aquí no existe el estado hover que sí tienen los botones web— y para cada

9 http://robots.thoughtbot.com/post/33427366406/designing-for-ios-taming-uibutton

FIG 11.6. En iOS las imágenes de botones se pueden diseñar previendo la forma en que crecerá el botón.



una de estas situaciones hace falta una imagen con variaciones visuales, que permita apreciar la diferencia.

Fondos

Los fondos no necesariamente tienen que ser imágenes que tengan el mismo tamaño que la pantalla para la que se diseña. Por ejemplo, un fondo con una textura o con patrones, puede ser solamente la imagen de un módulo del tamaño necesario que luego es repetido por código las veces que haga falta.

En el caso de los gradientes, se puede repetir varias veces una imagen que tenga el alto o el ancho necesario. Así, si hay un gradiente vertical, puede exportarse una imagen que tenga el alto de la pantalla por 1px de ancho y repetirla horizontalmente hasta que cubra la totalidad del espacio disponible.

En Windows Phone se presenta una situación particular: los diseños que se basan en Panorama tienen un formato sobre todo horizontal, abarcando varias pantallas[10].

10 Panorama control design guidelines for Windows Phone. http://msdn.microsoft.com/en-us/library/windowsphone/design/hh202912(v=vs.105).aspx

Íconos de pestañas y barras

Antes de crear íconos nuevos es recomendable revisar primero aquellos que ya vienen establecidos en cada sistema operativo.

En el caso de iOS, la fuente Apple Symbols incluye la mayoría de íconos que se usan habitualmente en las aplicaciones para iPhone o iPad y en Android, se puede descargar un pack de íconos oficiales[11]. Windows Phone también tiene una serie de íconos prediseñados[12].

Cuando sea necesario crear íconos por cuenta propia, es importante seguir la línea visual del sistema operativo con el objetivo de mantener una consistencia tal, que no dé oportunidad al usuario de diferenciar entre aquellos íconos que han sido diseñados especialmente para la app y los que no.

En Android, las imágenes para íconos de menú, pestañas y barra de acciones, tienen que diseñarse con el contraste necesario de acuerdo al tema —oscuro o claro— que se haya elegido al desarrollar la aplicación. En este caso, se trata de imágenes .png con fondo transparente.

Los íconos para diálogos deben tener el tratamiento gráfico de colores y sombras que recomienda Android en su guía[13] y los de pestañas deben incluir los estados correspondientes a «normal» y «seleccionado». Para mayor comodidad, existe una

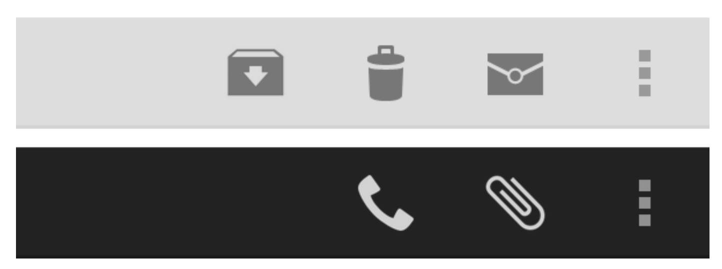


Fig 11.7. En Android los íconos tienen que diseñarse claros u oscuros para lograr el contraste suficiente con el fondo.

11 http://developer.android.com/downloads/design/Android_Design_Icons_20120814.zip
12 App bar icon buttons for Windows Phone. http://msdn.microsoft.com/en-us/library/windowsphone/develop/ff431806(v=vs.105).aspx
13 http://developer.android.com/design/style/iconography.html

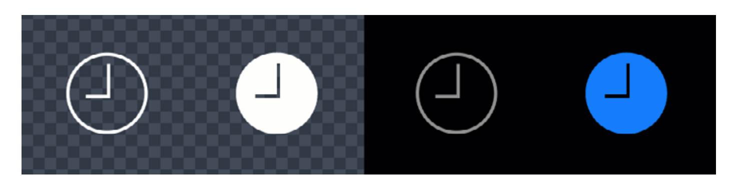


Fig 11.8. iOS añade efectos predefinidos a la imagen original del ícono –blanca y con fondo transparente–.

plantilla ofrecida por Android que sirve como guía y ayuda a imitar el estilo visual correspondiente a cada caso.

En iOS se aplican efectos para los íconos que se diseñan para la barra de herramientas, de navegación o de pestañas. Todas estas imágenes se usan como máscaras, por lo cual tienen que ser blancas, con antialias y con fondo transparente. También es importante no incluir sombras, y si se usan relieves, hay que asegurarse que estén a 90º, tal como si el ícono fuera iluminado por una luz por la parte superior.

Los efectos visuales predeterminados de los sistemas operativos pueden sobrescribirse si no se está conforme con ellos, siempre y cuando se provean las imágenes necesarias para reemplazarlos.

En el caso de Windows Phone, las nuevas imágenes creadas para íconos tienen que seguir la estética que propone el sistema operativo: minimalistas, formas planas, sin gradientes ni relieves. Es recomendable que estas sean blancas, ya que después tendrán color según el tema elegido por el usuario. En consecuencia, el archivo resultante será un `.svg` fácil de escalar.



Fig 11.9. En Windows Phone los íconos tienen formas simples y planas.

Nombres de archivos

Cuando se trabaja con muchas imágenes diferentes, sobre todo para distintas resoluciones, es importante mantener la organización y consistencia en la forma de nombrar los archivos exportados.

Trabajar de manera ordenada representa una ventaja a la hora de buscar imágenes dentro de un proyecto. Solo con el nombre debería ser posible identificar la función que cumple o el contexto donde se encuentra determinada imagen.

En Android se recomienda usar carpetas diferentes para cada una de las densidades. Allí se almacenarán las imágenes manteniendo el mismo nombre para cada una de los casos.

Como ayuda adicional a las carpetas, las imágenes pueden ser nombradas con prefijos. Por ejemplo, los íconos pueden empezar con `ic_`. Incluso, pueden sumarse combinaciones de prefijos para dar más claridad acerca del tipo de imagen. Un gráfico que se llame `ic_launcher_calendar.png` nos dará un indicio bastante preciso de qué es lo que hace y dónde se ubica.

Las imágenes que se exportan para iOS también tienen una forma especial de nombrarse dependiendo de si están destinadas para terminales con pantalla retina o no retina. En el primer caso hay que incluir `@2x` después del nombre. Por ejemplo, tendríamos la imagen `calendar@2x.png` para retina y `calendar.png` para no retina. Si se tratara de iPad, puede añadirse al final el sufijo `~ipad` para marcar diferencia con las imágenes para iPhone.

En Windows Phone, si se crean imágenes `.svg` escalables, no es necesario tener diferentes versiones. Sin embargo, al trabajar con imágenes `.png`, debería añadirse al final del nombre del archivo un sufijo que lo vincule con la densidad del dispositivo, como `_VGA`, `_WXVGA` o `_720p`, permitiendo indicar fácilmente qué versión se utilizará para cada caso.

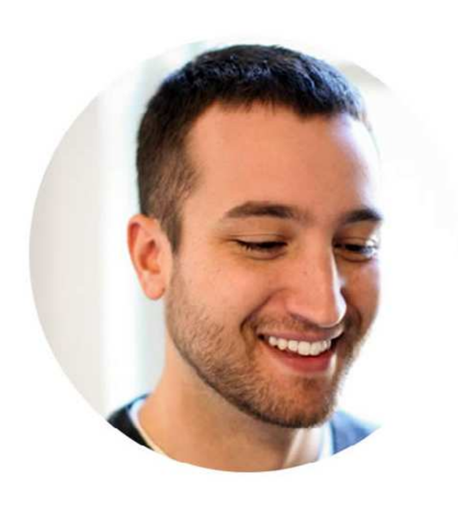
Comunicación con el desarrollador

Para asegurar una correcta implementación, diseñador y desarrollador deben mantener una comunicación constante y fluida.

Es posible que surjan dudas acerca del tamaño de gráficos o de fuentes, márgenes o uso de la retícula, imágenes que faltan o simplemente, cosas que el diseñador haya pasado por alto o no haya considerado oportunamente. Para despejar estas inquietudes, sobre todo cuando se trabaja de forma remota, es conveniente que el diseñador prepare una serie de documentos visuales que ayuden al desarrollador a interpretar los diseños.

Allí se pueden mostrar determinadas pantallas de la app —preferiblemente las más complejas o aquellas que más cambian entre sí— y, a través de líneas y demarcaciones, indicar los diferentes tamaños de tipografía y espacios entre los elementos visuales.

Si a pesar de esto, aún quedan preguntas sin resolver, lo ideal es estudiar cada situación de forma conjunta, decidiendo entre los dos la mejor solución.



12

Loren Brichter: Rompiendo todos los esquemas

En el mundo del diseño y desarrollo de apps hay pocos referentes. No son muchas las personas que han «pateado el tablero» para salirse de los esquemas y proponer conceptos realmente novedosos. Loren Brichter es una de ellas: no por nada fue llamado «el Sumo Sacerdote del diseño de apps» por el Wall Street Journal[1].

Aunque ni siquiera alcanza los 30 años, Loren ya tiene una carrera digna de envidiar. Hace unos años pasó por Apple, donde ayudó a diseñar el software que llevó el primer iPhone, y él mismo nos cuenta su aporte a ese proyecto:

> *Mi trabajo fue muy abajo en la pila de software, escribiendo el código gráfico de bajo nivel, lo cual permitió que mecanismos de interfaz de más alto nivel pudieran funcionar. Fue un honor formar parte de ese proyecto.*

Apple no fue la única empresa grande donde trabajó; después de su paso por la compañía de la manzana, formó parte de la plantilla de Twitter, a donde llegó impulsado por el éxito que había cosechado con Tweetie, un proyecto personal en el que incorporaba algunos elementos de diseño que llamaron la atención de la red de microblogging.

Actualmente, Brichter trabaja desde Filadelfia en Atebits, su propia empresa, alejado del epicentro de la tecnología en nuestros días, Silicon Valley.

Has trabajado tanto en grandes empresas como por tu cuenta, en Atebis. ¿Qué has aprendido de esta experiencia?

> *Aunque he trabajado para dos compañías increíbles, Apple y Twitter, creo que disfruto más trabajando para mí mismo. Esto requiere un tipo de disciplina diferente y por*

1 Lessin, Jessica E. High Priest of App Design, at Home in Philly. http://online.wsj.com/article/SB10001424127887324392804578358730990873670.html

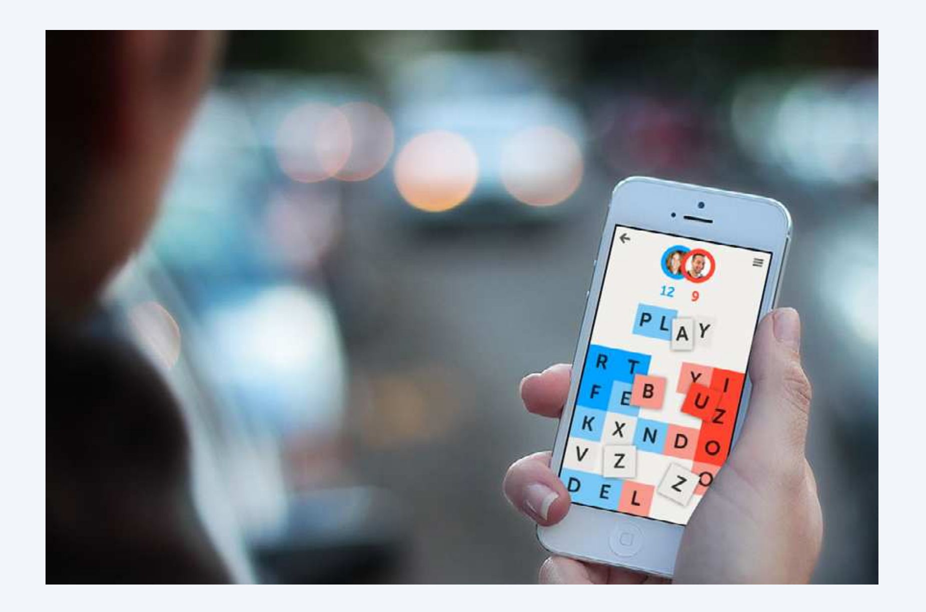


FIG 12.1. *El estilo visual de Letterpress se caracteriza por su limpieza, resaltando los elementos más importantes.*

> *supuesto, echo de menos algunos recursos que las grandes empresas pueden proveer; sin embargo, ¡es emocionante trabajar en algo sobre lo cual tienes completa libertad!*

La última aplicación de Loren se llama Letterpress, un juego en línea que permite formar palabras usando letras definidas al azar, desafiando a oponentes en cualquier lugar del mundo. La aplicación ha tenido tanto éxito, que ha sido usada por Apple como ejemplo en las promociones del iPad mini.

Esto se debe tanto a la experiencia de uso, como a la sana adicción que ha generado en los usuarios, el famoso engagement que muchas aplicaciones buscan conseguir. Su interfaz es bastante limpia, basada en el blanco y poniendo a los bloques de letra, auténticos protagonistas del juego, como centro de atención.

Al momento de lanzarse Letterpress, su diseño de interfaz no estaba demasiado apegado a la personalidad de un iOS en el que, por entonces, predominaban relieves, sombras y texturas.

Por esta razón, le preguntamos si piensa que este estilo más independiente y abstraído de la estética de la plataforma puede ser una tendencia que otras aplicaciones imiten en el futuro:

> *El mercado es relativamente joven y estamos empezando a ver cómo surgen algunas convenciones, pero la innovación aún se está extendiendo por todos lados. Por mi parte, estoy tratando de identificar buenos patrones para extrapolarlos; descubriendo nuevas e interesantes maneras de interactuar con los dispositivos.*

Estamos seguros de que tienes un montón de ideas de apps en la cabeza. ¿Cómo te das cuenta cuando una de ellas es lo suficientemente madura como para convertirse en un proyecto?

> *Sería como el arte versus porno, simplemente te das cuenta cuando lo ves. Hay grados de maduración. He tenido algunas ideas con el potencial para ser productos viables, pero Letterpress parecía tener un buen conjunto de atributos que harían divertido lanzarla al mundo en ese momento.*

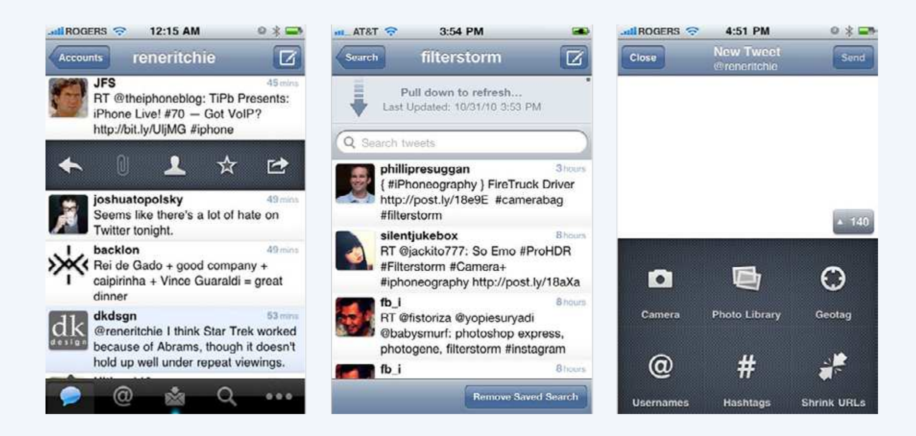


FIG 12.2. *Tweetie incorporaba varios mecanismos de interacción inventados por Brichter que luego otras apps comenzaron a imitar.*

Dos consideraciones importantes son: ¿otras personas la encontrarán divertida y útil? ¿puede ser finalizada? Es muy difícil marcar un límite en la versión 1.0. Es importante que la idea pueda llegar hasta ese punto.

Loren Brichter es también el creador de «soltar para actualizar», una función que se ha masificado y ha llegado a utilizarse en muchas aplicaciones, incluidas Facebook, Path y Pinterest, entre otras. Lo mismo sucedió con «deslizar sobre una fila» para mostrar más opciones o los paneles laterales, que también inventó.

Como esos, hay muchos otros mecanismos de interacción que, aunque no han salido de la cabeza de Loren, están siendo usados constantemente; un ejemplo son las listas de navegación a las que se llega a través del famoso «botón hamburguesa».

Desde una perspectiva de alto nivel, es divertido ver cómo ideas como esa se diseminan y evolucionan en el ecosistema. Honestamente, soy un fan del panel de navegación apilada: convierte un concepto abstracto en algo físico y real. Estoy seguro de que veremos muchos más experimentos siguiendo esta misma línea, diseños que aprovechen nuestra expectativa de realidad física para hacer interfaces más intuitivas.

A todas luces, hablamos de un personaje muy inquieto, que constantemente está pensando en nuevas ideas y eligiendo unas pocas que se convertirán, según refleja su historial, en productos exitosos.

Si quieres saber más sobre Letterpress, puedes encontrar información en: **www.atebits.com**



13

Ejemplos a seguir

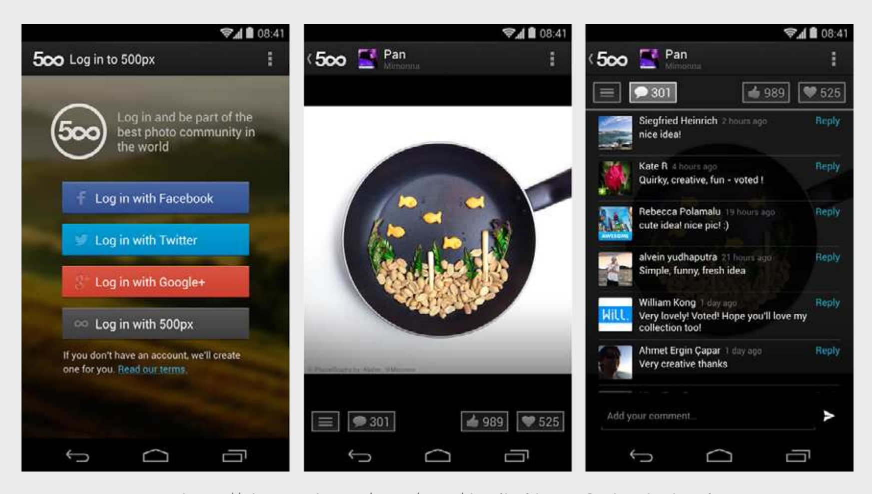
Como vale la pena aprender de aquellos que ya recorrieron el camino, hemos seleccionado tres apps de cada sistema operativo que para nosotros, por diferentes motivos, han hecho las cosas bien.

ANDROID

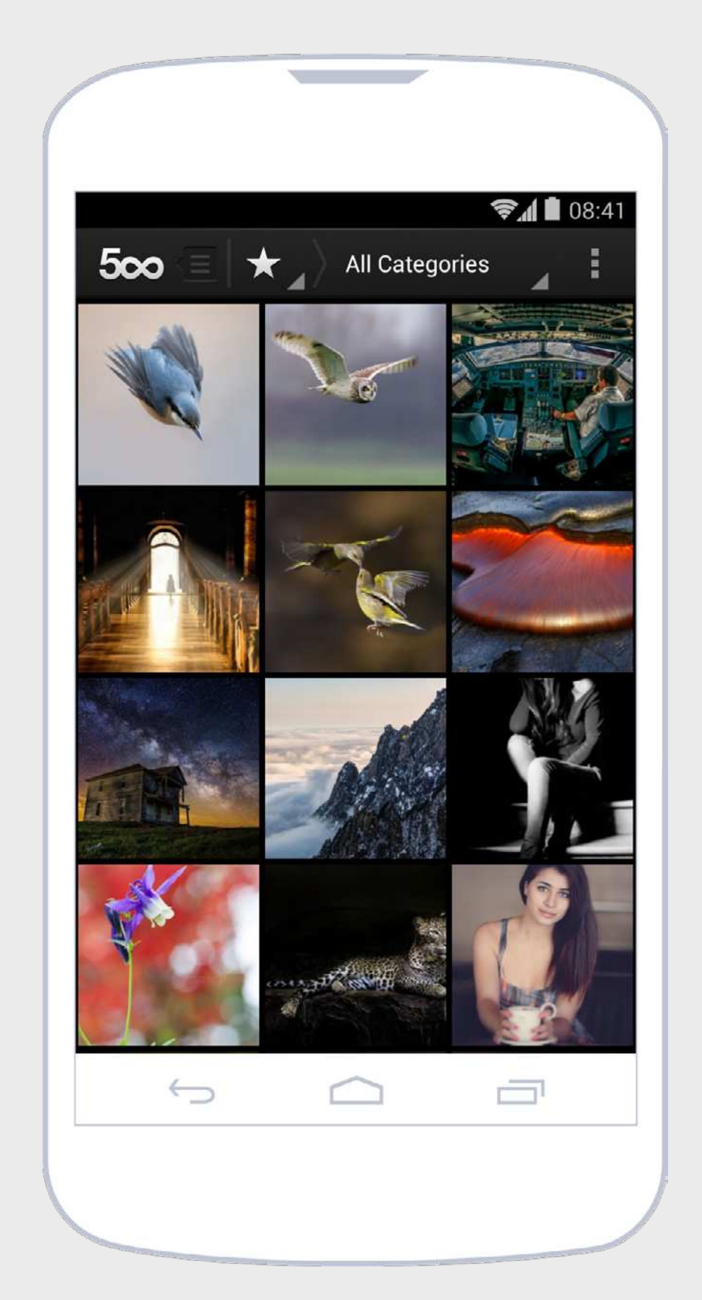


500px[1]

La app de la comunidad de fotógrafos hizo bien los deberes. Además de ofrecer múltiples medios de registro y de manera clara, ha sabido trasladar la cualidad de «exploración» de esta plataforma. Visualmente es una aplicación muy cuidada, donde el uso de colores oscuros para los controles ayuda a realzar las fotografías.



1 https://play.google.com/store/apps/details?id=com.fivehundredpx.viewer



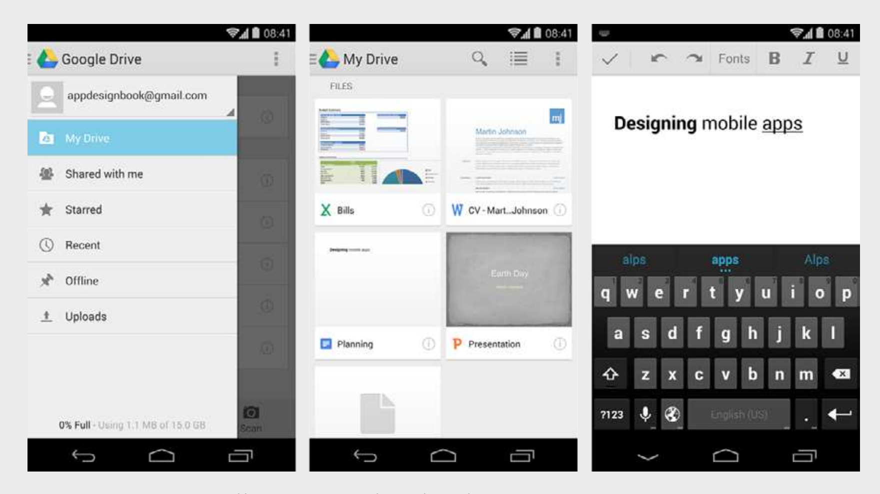
← *Fig 13.1.* *Pantallas de acceso, de detalle de una foto y de comentarios*

Fig 13.2. *Pantalla de categorías.*



Google Drive[2]

Este servicio de almacenamiento y colaboración de documentos «en la nube» se centra en sus funciones principales, pero también ofrece características de edición de documentos, ideales para usuarios que cambian frecuentemente entre el ordenador de escritorio y el móvil, que no quieren perder esta capacidad.



2 https://play.google.com/store/apps/details?id=com.google.android.apps.docs

← *FIG 13.3. Menú tipo cajón, vista previa de archivos y edición de un documento.*

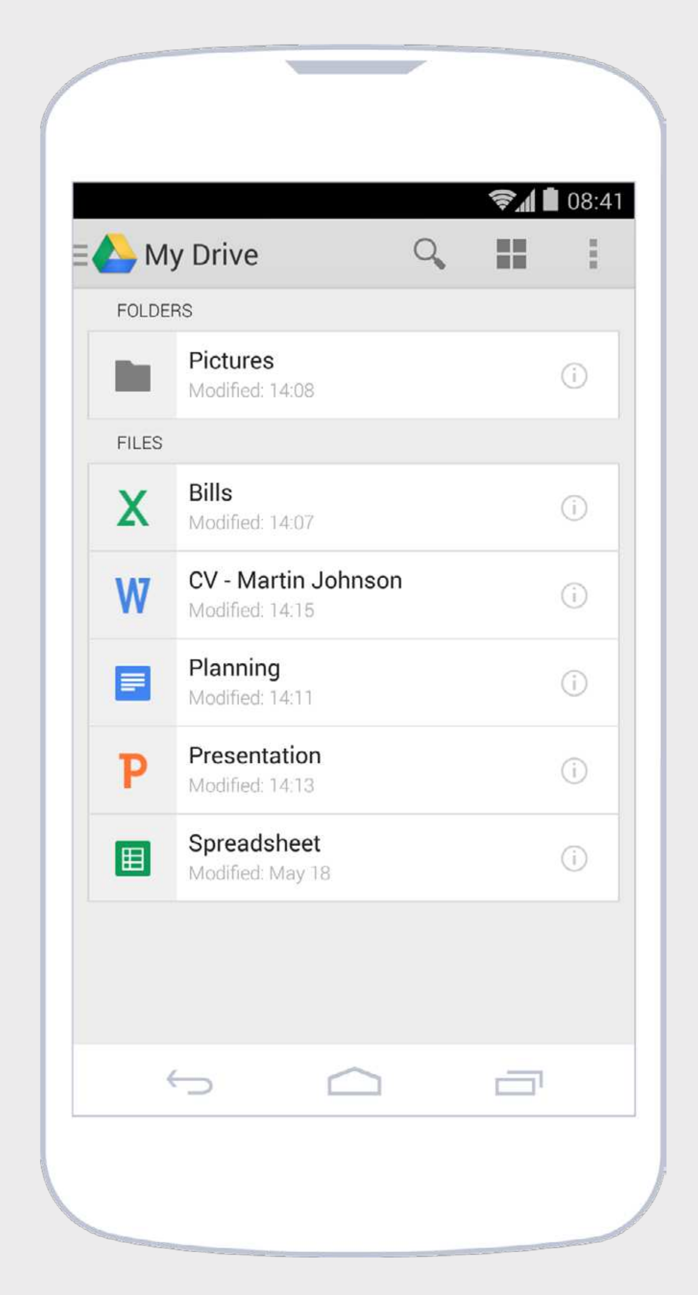
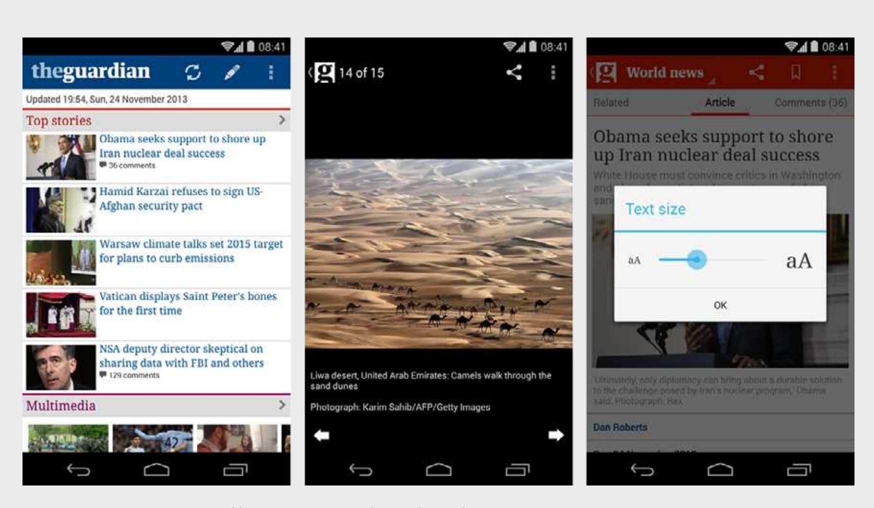


FIG 13.4. Pantalla de lista de documentos.



The Guardian[3]

El diario inglés hace uso extensivo de elementos nativos y, a la vez, aplica al pie de la letra su identidad corporativa, utilizando las cabeceras como el elemento más fuerte para comunicarla. Por otro lado, establece claras jerarquías en textos de lectura y no se olvida de añadir características de accesibilidad como el aumento del tamaño de las fuentes.



3 https://play.google.com/store/apps/details?id=com.guardian

← *FIG 13.5. Pantalla principal de noticias, detalle de una foto y configuración de tamaño de texto.*

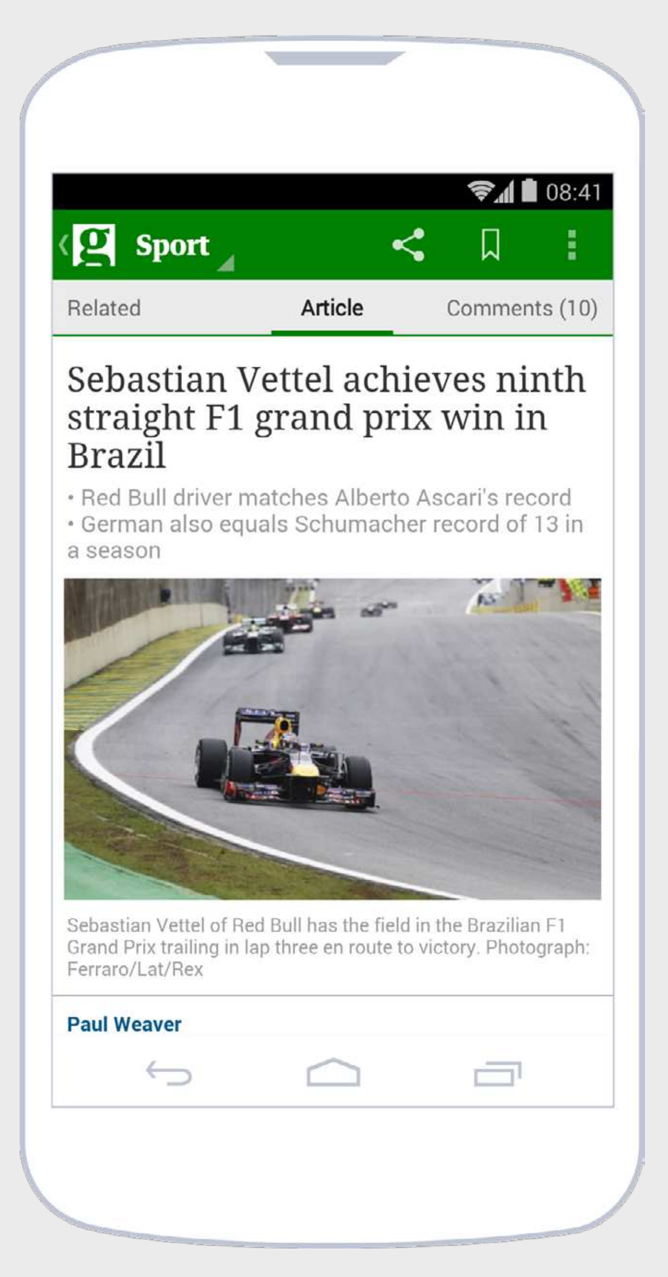


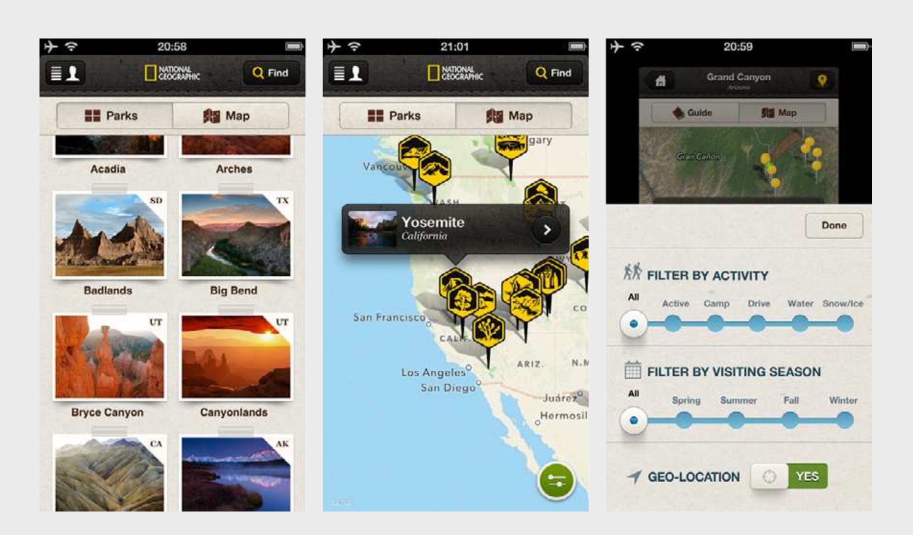
FIG 13.6. Detalle de una noticia.

iOS



National Parks[4]

Tanto en iPhone como en iPad, la atención a los detalles de esta app de National Geographic es asombrosa. A lo largo de las pantallas, puede apreciarse un uso de texturas justificado, con transiciones que aportan valor al concepto. Por otro lado, las jerarquías en la composición son muy claras, lo que favorece la interpretación de los contenidos.



4 https://itunes.apple.com/us/app/national-parks-by-national/id518426085?mt=8&uo=4

← *Fig 13.7.*
Parques vistos como fotografías, como ubicaciones en un mapa, opciones de filtrado.

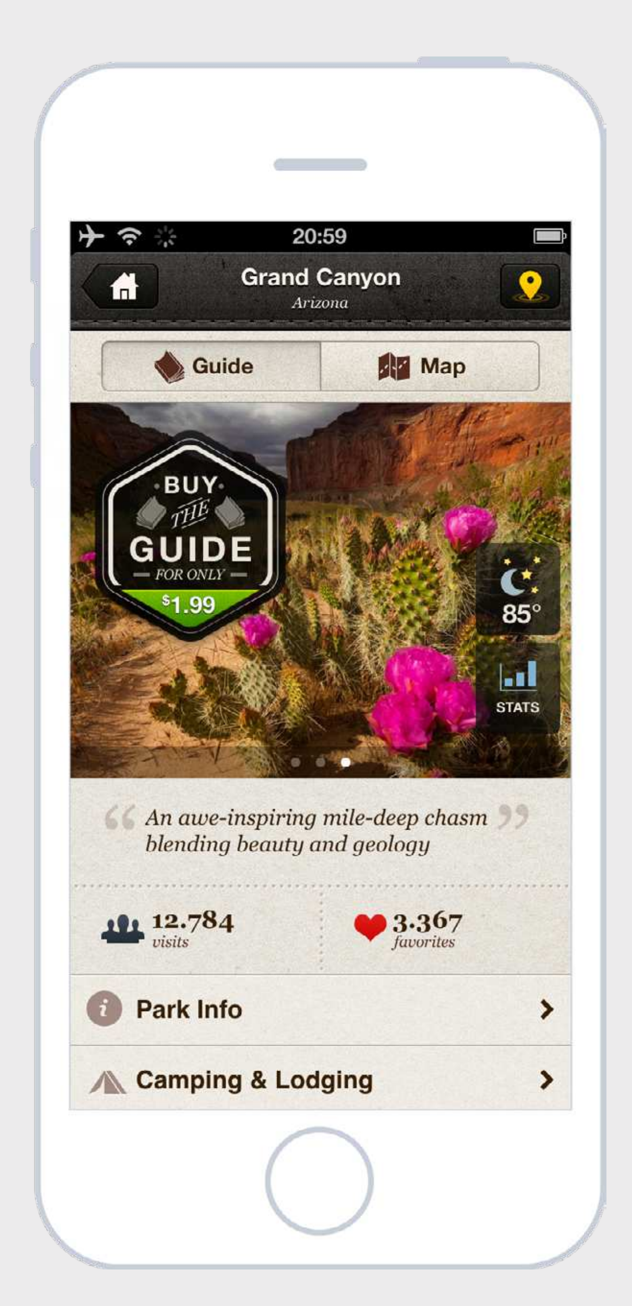
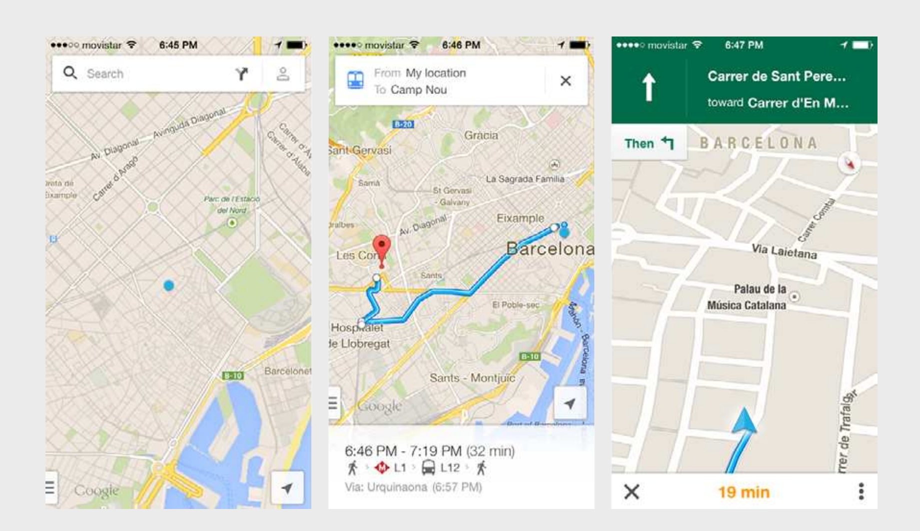


Fig 13.8.
Detalle de un parque.

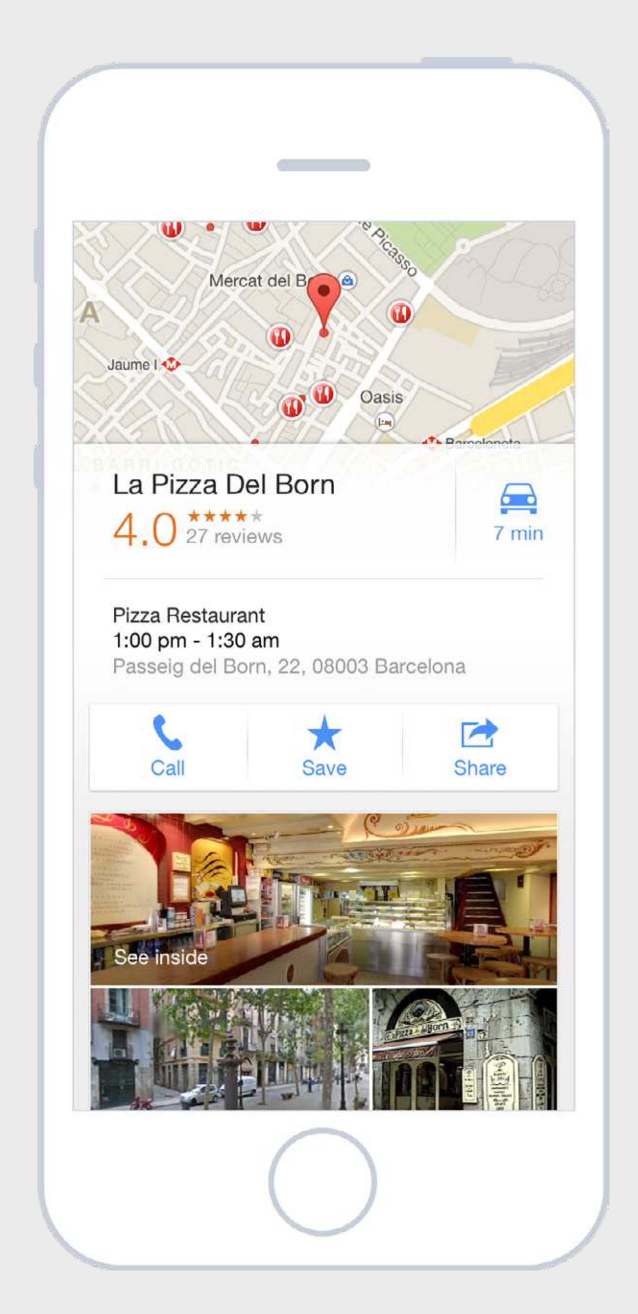


Google Maps[5]

La aplicación que todos los usuarios de iPhone esperaron durante meses no defraudó. Desde el lanzamiento de esta app, Google ha demostrado su intención de establecer sus propias normas allí donde va. Nos gustó porque hace un uso consciente del espacio, en cada pantalla dispone los elementos en relación directa a las necesidades temporales, ocultando y destacando lo justo y necesario. Simple y robusta a la vez.



5 https://itunes.apple.com/us/app/google-maps/id585027354



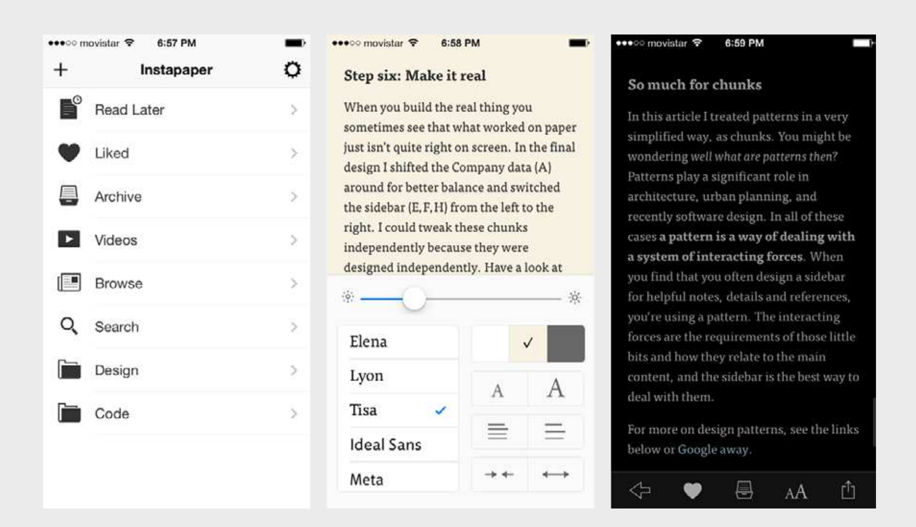
← *FIG 13.9. Pantalla principal de búsqueda, resultado con recorrido y pantalla de navegación.*

FIG 13.10. Detalle de resultado con opciones.

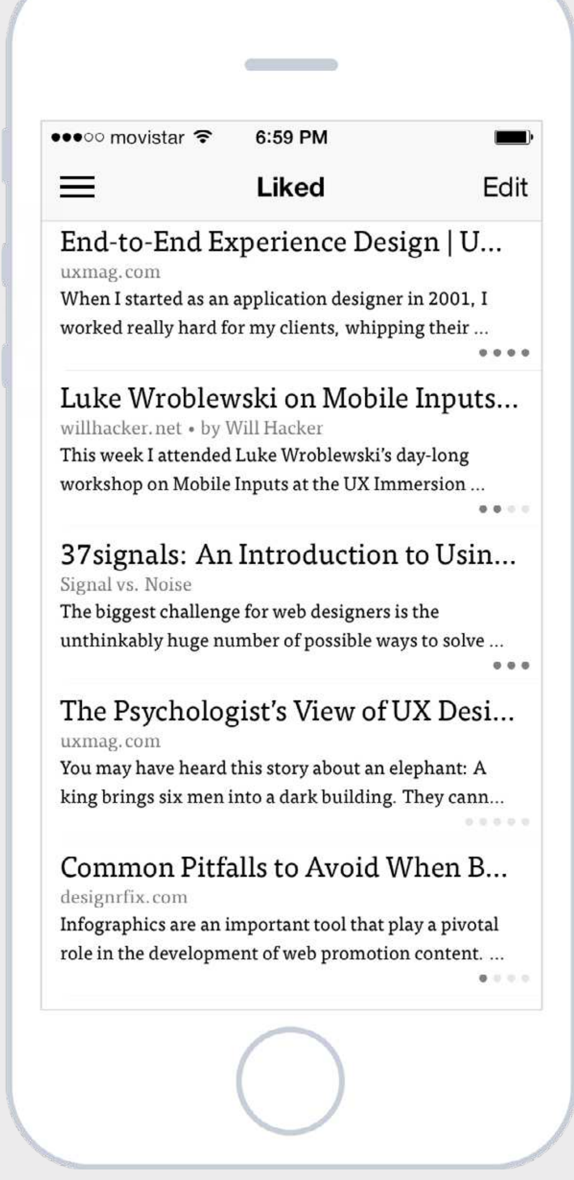


Instapaper[6]

Una de las pioneras alrededor de la idea de «leer después», la app de lectura de textos de Marco Arment está más que pulida y en ella todo gira en torno a su tarea principal: facilitar la lectura. Flexible en todo momento, es capaz de ocultar controles en situaciones que no los requieren o cambiar a «modo nocturno» para adaptarse a las condiciones de luz.



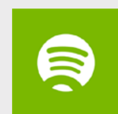
6 https://itunes.apple.com/us/app/instapaper/id288545208



← *FIG 13.11.* *Menú principal, pantalla con opciones de lectura y en modo de lectura nocturna.*

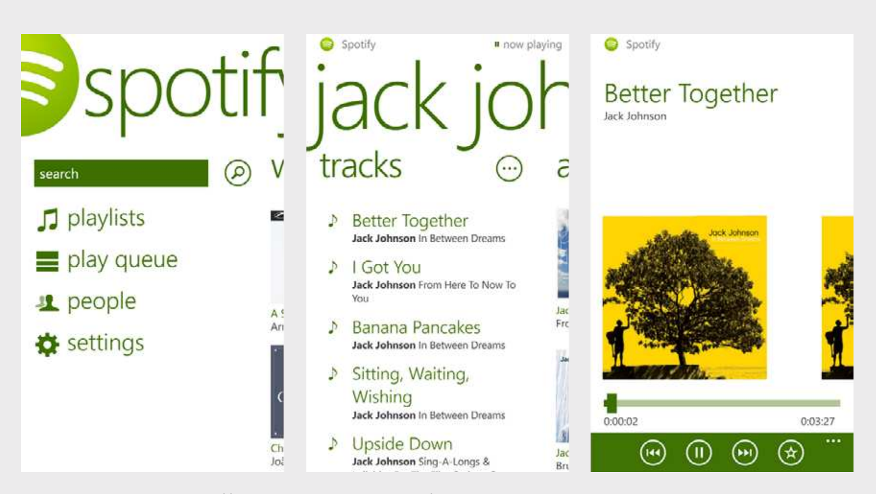
FIG 13.12. *Lista de ítems marcados como favoritos.*

Windows Phone



Spotify[7]

El servicio de música por Internet ha sabido adaptarse a la propuesta de Windows Phone. Aplica su identidad sobre patrones como *Panorama* y *Pivot Control* como pocas. Un excelente equilibrio entre funciones, identidad y consistencia con el sistema operativo.

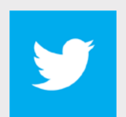


7 http://www.windowsphone.com/s?appid=10f2995d-1f82-4203-b7fa-46ddbd07a6e6

← *FIG 13.13. Pantalla principal con Panorama, lista de canciones y opciones de reproducción de canción.*

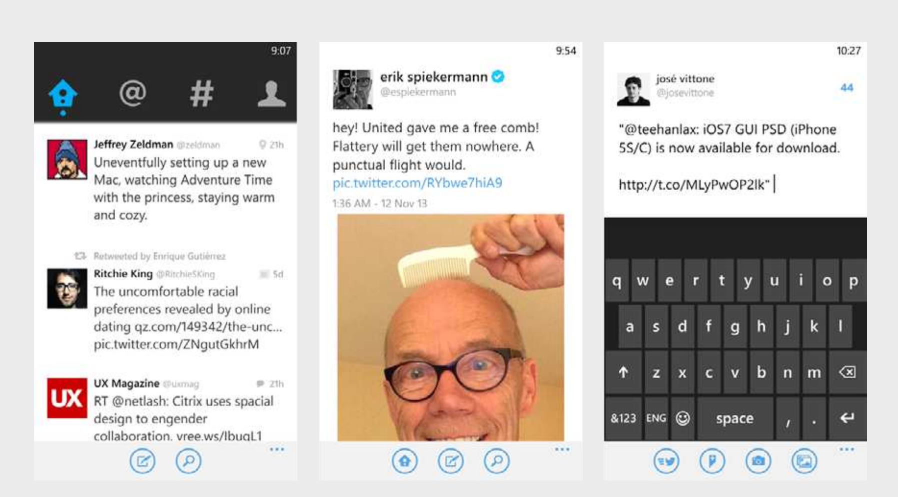


FIG 13.14. Pantalla de álbumes de un artista.

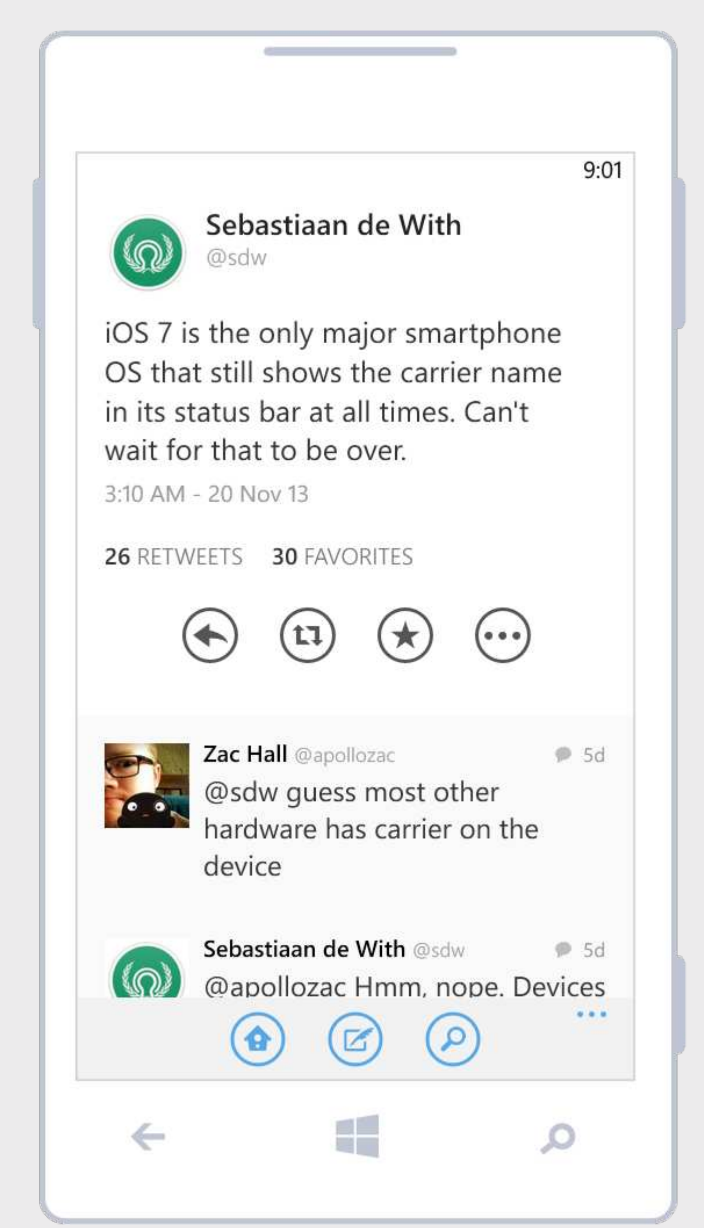


Twitter[8]

De un vistazo podemos saber que se trata de la app de Twitter y eso no es poco. Han sabido dominar los puntos de contacto entre elementos nativos e identidad visual, sin perder de vista las guías propuestas por Microsoft. Un claro ejemplo de una aplicación centrada en el contenido.

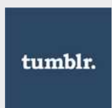


8 http://www.windowsphone.com/s?appid=0b792c7c-14dc-df11-a844-00237de2db9e



← *FIG 13.15. Pantalla principal, detalle de un tweet con foto y de composición.*

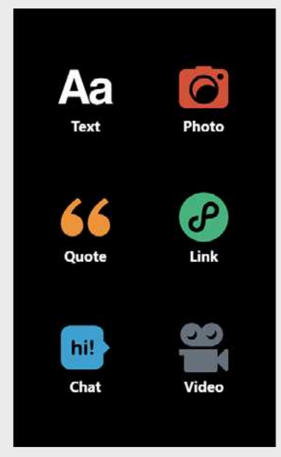
FIG 13.16. Detalle de un tweet.

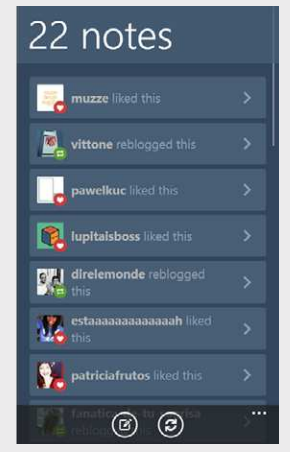


Tumblr[9]

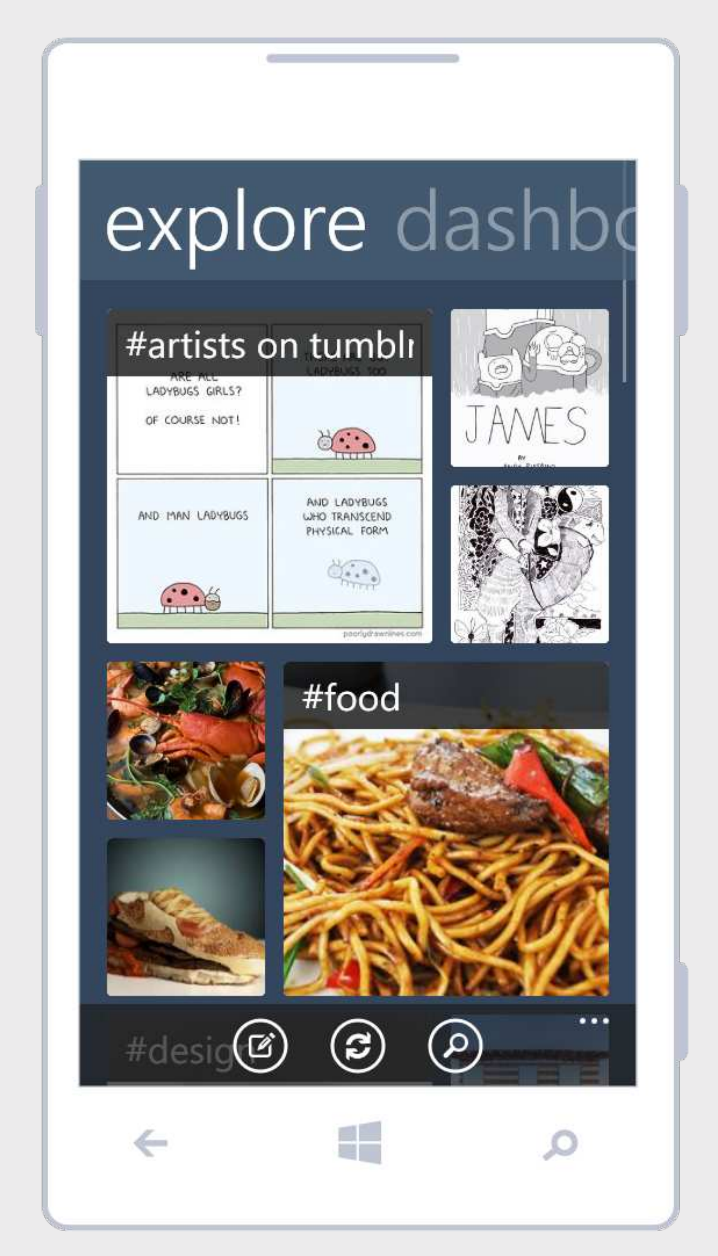
La plataforma de *microblogging* ha comprendido el entorno móvil y ha respondido con una app simple y eficaz. Manteniendo funciones conocidas por sus usuarios web, la aplicación hace foco en la exploración de material visual y la publicación de contenidos rápidamente.







9 http://www.windowsphone.com/s?appid=ffa2fb4f-61b2-4075-ac7b-488846998b72



← *FIG 13.17. Pantalla de inicio, selección de tipo de entrada nueva y nota de una entrada.*

FIG 13.18. Exploración de contenidos.



14

El mundo tableta

Diseñar para tabletas implica mayor responsabilidad para aprovechar el espacio y las circunstancias disponibles. Pasar una aplicación del móvil a su hermana mayor no debe ser una traslación literal, sino medida y con sentido.

Los usuarios alternan constantemente entre un dispositivo y otro. La mayoría de las veces que una persona está usando una tableta, al mismo tiempo, está prestando atención al móvil o al televisor.

Por esta razón, si ya se tiene una app para móvil, el siguiente paso natural es diseñar una versión para tablet; incluso, podría decirse que en algunos casos, esto es una prolongación obligada de la experiencia.

Aunque a primera vista parecen similares, en realidad se trata de dos medios que se diferencian, no solo en tamaño, sino también en los modos de uso —cómo y dónde son utilizados— y en la relación que generan con el usuario.

En este sentido, tanto el diseño visual como de interacción se ven particularmente afectados por las características de la tableta, por lo cual, comprenderlas permite sacar mejor provecho de ellas.

Llegando a la pantalla grande

Tener tanto espacio disponible de repente, puede desorientar al principio. Como también pasa en el caso del móvil, el éxito del diseño de una app para tableta radica, en gran parte, en un correcto aprovechamiento del espacio.

La pantalla de una tableta no se debe interpretar solo como un área más grande en la cual se puede volcar directamente la interfaz de la aplicación diseñada para móvil. Entre los aspectos a considerar, se encuentran:

Mayor protagonismo visual

Una tableta permite apreciar mejor los detalles gráficos y el tratamiento de una interfaz. La pantalla del móvil limita el espacio de forma tal que los componentes visuales están bas-

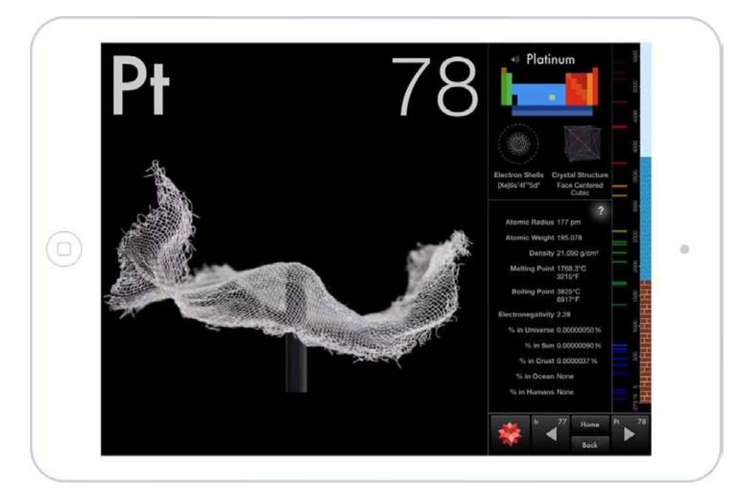


FIG 14.1. El espacio disponible en una tableta permite que los gráficos se luzcan más en pantalla.

tante condicionados por el área que los rodea. En cambio, en la tableta, algunos íconos e ilustraciones tienen mayor espacio para mostrarse y lucirse.

Lo anterior, afecta directamente la calidad de las imágenes y el nivel de detalle con el que se trabaja. Detalles que antes no tenían sentido en el móvil porque no se apreciarían, en la tableta encuentran un lugar donde ser protagonistas, dando mayor valor al diseño de la interfaz.

Tipografía

Comparadas con el móvil, las tabletas se sostienen de una forma en que la pantalla se encuentra a mayor distancia del ojo del usuario. Esto obliga a adaptar especialmente las fuentes utilizadas y sus características visuales, para amoldarlas a la nueva situación de uso y mantener así la legibilidad de los textos. Entonces, mientras más alejada esté la pantalla del usuario, más grande debería ser la fuente y mayor el interlineado para asegurar una correcta legibilidad.

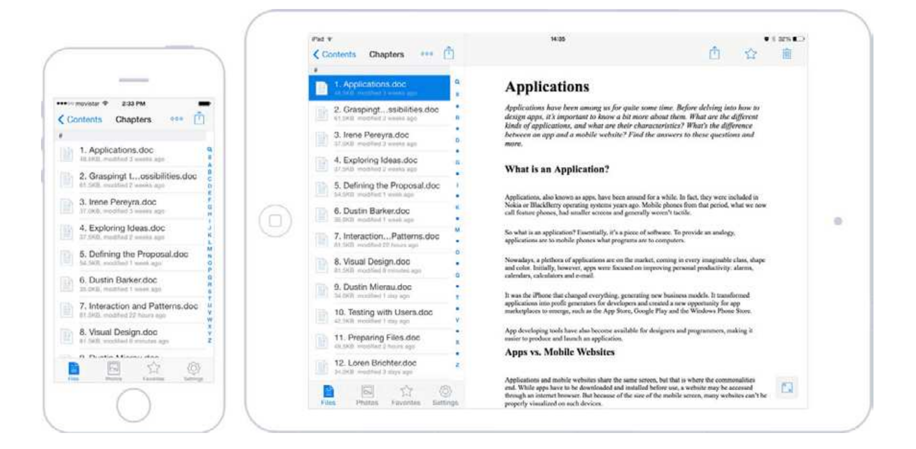


Fig 14.2.
Las tabletas hacen un uso diferente del espacio, lo que permite mostrar dos pantallas en una sola. En este caso, Dropbox muestra la lista de archivos junto con el contenido, algo que no se puede hacer en iPhone sin cambiar de pantalla.

Aprovechamiento del espacio

La forma en que se gestiona un espacio mayor no incide solamente en el aspecto visual, sino también en la navegación y, por consiguiente, en la interacción entre el usuario y la interfaz.

Mientras en el móvil el espacio reducido obliga a realizar una mayor cantidad de pasos para llegar a la información, en una tableta esta secuencia se puede reducir. Ahora se cuenta con un espacio más amplio que permite, por ejemplo, incluir el equivalente a la información contenida en dos pantallas de móvil separadas, en una sola vista integrada.

Para sacar partido de una situación como esta y para poner un ejemplo de uso, puede usarse un panel lateral izquierdo que incluya un listado de ítems, cuyo contenido se mostrará en la zona derecha de la pantalla al mismo tiempo. De esta forma, elemento y contenido se relacionan en un solo vistazo sin cambiar de contexto. En una aplicación de correo, es posible mostrar el listado de los elementos de una bandeja de entrada y

el contenido del mensaje sin necesidad de cambiar de pantalla, como ocurriría en el caso de un teléfono.

Un mayor espacio también beneficia la navegación, ya que no se necesitan tantos niveles para llegar a un contenido específico. Pantallas que antes se encontraban separadas en distinta profundidad, ahora pueden compartir el mismo nivel.

Los sistemas operativos proponen diferentes formas de aprovechar el espacio y gestionar la división entre paneles. Independientemente de la forma que se elija, es importante relacionar el ítem seleccionado con el visualizado y disminuir la interrupción provocada al usar una navegación entre pantallas diferentes. Esto se consigue llevando el contenido principal hacia una zona central, apoyándose en el uso de diálogos o elementos flotantes para acciones secundarias.

Estas vistas divididas deben considerarse tanto en formato horizontal como vertical y debe preverse cómo se adaptarán estos paneles a los cambios de orientación. Es importante considerar esto último, teniendo en cuenta que en una tableta el cambio de orientación es mucho más frecuente que en un móvil. En iPad, por ejemplo, elementos que se encuentran en la columna izquierda en formato horizontal, pueden ser accesibles desde un menú desplegable cuando la tableta está en posición vertical.

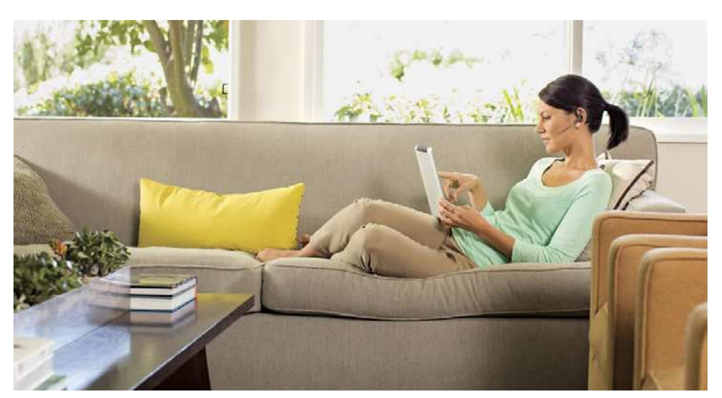


FIG 14.3. El contexto de uso de las tabletas está asociado al hogar, donde los usuarios están más relajados y con más tiempo disponible.

Los gestos

A veces en un móvil resulta incómodo realizar aquellos gestos que incluyen más de dos dedos a la vez, por el poco espacio disponible y la distancia que los separa al realizar los movimientos. En una tableta los gestos pueden realizarse de forma más fácil.

Por este motivo, aquí hay más libertad a la hora de manipular directamente los elementos gráficos de la interfaz —sin necesidad de basarse tanto en botones y controles— para llevar adelante algunas acciones. Por ejemplo, rotar una fotografía que antes podía ser algo dificultoso, ahora puede apoyarse en el uso de dos o tres dedos y conseguirse más fácilmente.

Contexto de uso

El dispositivo que utilizamos en general está afectado por nuestro contexto: dónde estamos, qué queremos hacer y cuánto tiempo tenemos para ello.

Una tableta suele estar más relacionada con el ámbito hogareño. Es normal usarla entonces cuando se está recostado en el sofá o en la cama. Por esto, el usuario tiende a estar más relajado y posiblemente con más tiempo disponible que cuando usa el móvil.

Como dato estadístico, podemos mencionar que los usuarios de tableta invierten casi el doble de tiempo en Internet que los usuarios de teléfono, especialmente para mantenerse informados y entretenidos[1].

Este contexto es una diferencia de mayor importancia porque afecta la relación del usuario con la tableta y la forma que tiene de interactuar con ella y, en consecuencia, de usar las aplicaciones.

1 http://services.google.com/fh/files/misc/multiscreenworld_final.pdf

La interacción

Una tableta es un objeto de tamaño mayor que un móvil y por consiguiente, es más pesado y difícil de transportar. Así, el usuario necesita un punto de apoyo que le permita soportar el peso de la tablet.

De acuerdo a esta forma de sostenerla, tanto en formato horizontal como vertical, las zonas a las que el usuario puede llegar más cómodamente son las esquinas ubicadas en la zona inferior de la pantalla. Aquí deberían estar las acciones más frecuentes y fuera de esta zona de comodidad se recomienda situar los controles de las acciones más críticas o que ocasionan mayor impacto, para hacerlas más difíciles de acceder, disminuyendo el riesgo de ser pulsadas por equivocación.

Por otro lado, también deberían reducirse los elementos interactivos localizados en el centro superior de la pantalla. Ubicarlos en esta área requiere que la mano y parte del brazo se crucen sobre la pantalla, tapando parte del contenido. Si esto pasara, podría no verse claramente el elemento que se está pulsando y el resultado de la acción.

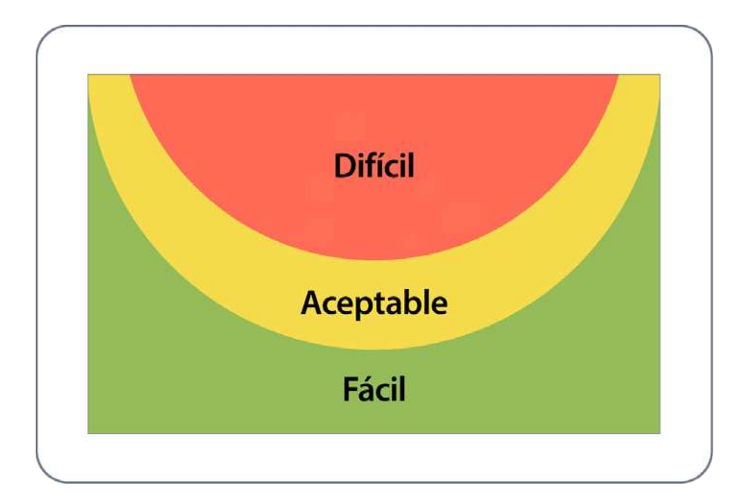


FIG 14.4. Según la manera de sostener la tableta, las zonas inferiores de la pantalla son más fáciles de acceder.



FIG 14.5. Spotify es un ejemplo de aplicación muy bien resuelta en su versión para iPad.

La funcionalidad

Que un usuario esté en casa tranquilo y que la tableta sea un dispositivo no tan personal como el móvil —ya que se comparte más fácilmente—, obliga también a repensar aquellas tareas que pueden llevarse a cabo con la aplicación.

En un móvil, para un usuario que muchas veces está en movimiento y más centrado en ejecutar tareas rápidas, algunas funciones no tienen razón de ser. En el caso de una tableta, la situación es distinta en cuanto al tiempo y a la comodidad, por eso, algunas características funcionales que se habían dejado postergadas ahora encuentran su razón de ser.

En este caso, es posible ampliar y mejorar la experiencia de uso sin perder el foco de atención de aquellas tareas que contribuyen al objetivo de la aplicación. Como comentamos antes, se trata de dar sentido y no de «incluir por incluir».

Un ejemplo de una app que potencie la experiencia, podría ser aquella donde se visualicen documentos, pero para editarlos haya que hacer uso de formularios. Mientras en un móvil esta tarea puede ser bastante difícil, en una tableta el ingreso de datos y texto puede ejecutarse más fácilmente.

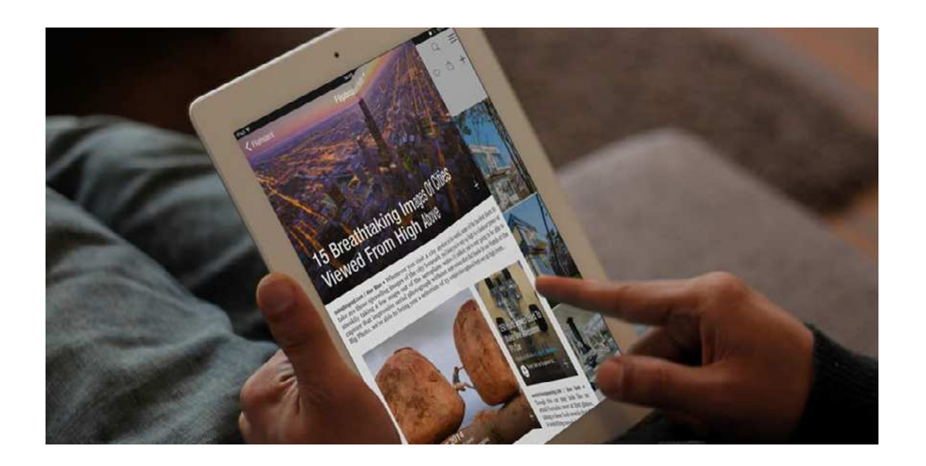


FIG 14.6. Flipboard se asemeja a una revista y refuerza este concepto mediante transiciones.

Contenido de la app

Las tabletas son el blanco preferido de algunas aplicaciones que por su tipo de contenido necesitan un contexto de uso más relajado y tranquilo, como cuando el usuario está en casa.

Este es el caso de apps que requieren la lectura de textos largos como periódicos y revistas. Un ejemplo de esto es Flipboard en su versión para iPad[2], una aplicación que realmente saca partido de la experiencia al asemejarse a una revista, trasladando el concepto tanto a la cantidad de información, como al diseño visual y a las animaciones.

2 http://flipboard.com/



15

Erik Spiekermann: El Maestro de la tipografía

Este señor alemán es una eminencia, si de tipografía se habla. En su extensa carrera ha diseñado gran cantidad de fuentes tipográficas, entre las que podemos destacar ITC Officina y FF Meta; esta última, llamada por algunos diseñadores, «la Helvética de los 90».

Meta, basada en formas más redondas que Helvética, ha sido usada en infinidad de diseños alrededor del mundo. Firefox no fue la excepción y la eligió como parte de su identidad corporativa. Erik ha trabajando conjuntamente con ellos para crear Fira[1], una tipografía basada en Meta, más simple y ancha para hacerla más legible en pantalla, ideal para el recientemente estrenado sistema operativo de Firefox para móviles, Firefox OS.

Spiekermann explica que los entornos digitales no suponen una condición nueva para la tipografía, sino que pueden compararse con escenarios particulares que existen tiempo atrás:

> *Mala resolución, tipografía pequeña, montones de información, lectores apurados y ambientes cambiantes… estas no son condiciones nuevas y los diseñadores de información han estado lidiando con ellas desde siempre. En lo que concierne a la tipografía, la pantalla es solamente papel de mala calidad.*

Sin embargo, no es una tarea simple elegir la tipografía para una app. No basta con buscar aquellas que mejor desempeño tienen en papel impreso de mala calidad, existen más parámetros a tener en cuenta. Afortunadamente, Erik viene al rescate y nos dice lo que deberíamos tener en cuenta a la hora de la elección tipográfica:

> *Considera las fuentes tipográficas que fueron diseñadas para información. Si no hay problemas de espacio —y generalmente los hay— usa aquellas que funcionan*

1 http://www.mozilla.org/en-US/styleguide/products/firefox-os/typeface/

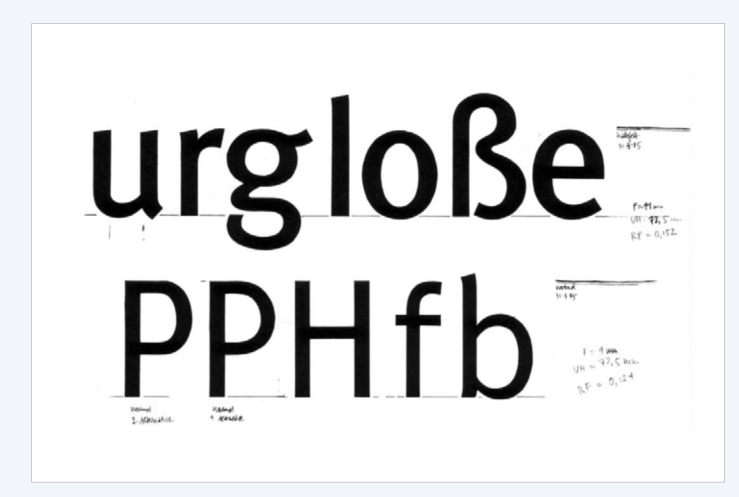
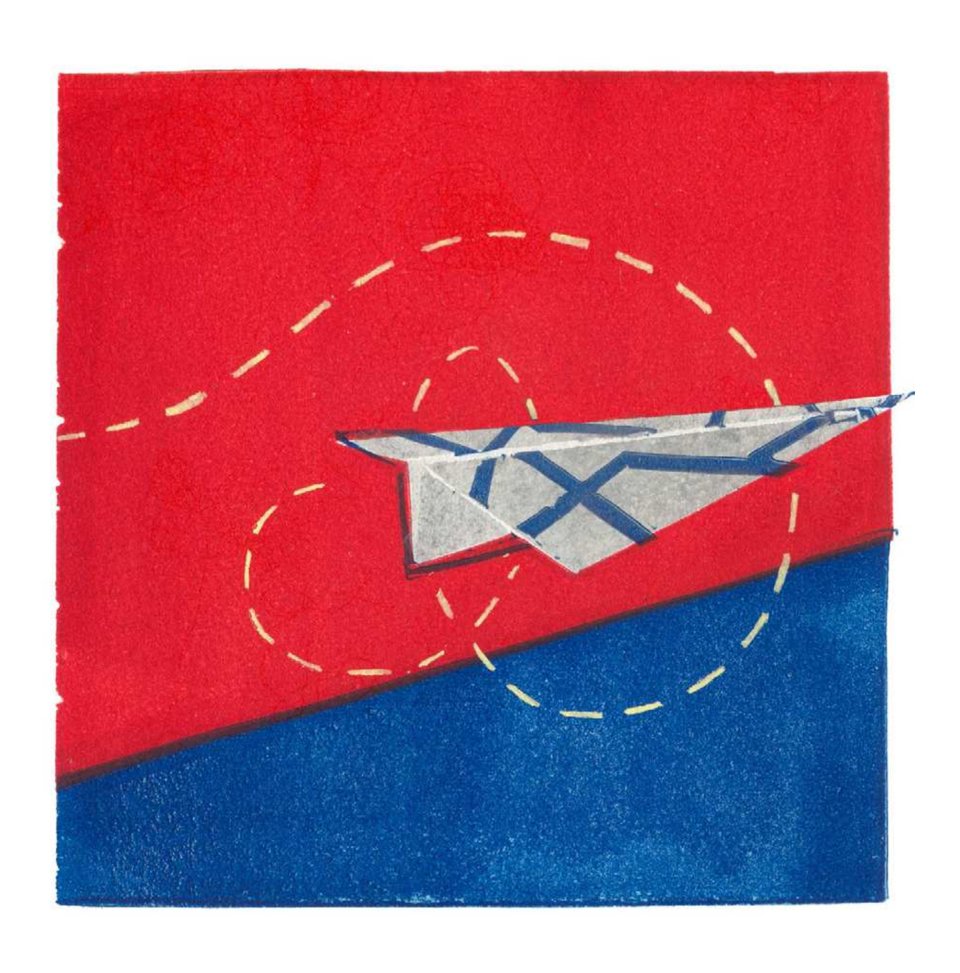


Fig 15.1. Meta se ha convertido en un clásico moderno, usada en todo el mundo.

> *para guías telefónicas o formularios. Todas mis fuentes lo hacen, por ejemplo, FF Info, FF Unit, FF Meta, ITC Officina; también Bell Gothic, Bell Centennial, Lucida —aún sigue siendo la mejor de Apple— más un montón de nuevas y robustas tipografías como FF Tisa, FF Yoga, FF Suhmo. Bajo contraste para las fuentes serif, formas abiertas —no como Helvética, pero sí como Frutiger— y algo de contraste para las fuentes sans-serif.*

Si lo dice Spiekermann, con años de experiencia en el diseño tipográfico, es mejor escucharlo.

Actualmente, Erik forma parte de Edenspiekermann, una agencia de comunicación y diseño con oficinas en Amsterdam, Berlín, Stuttgart y San Francisco. Visita su web en www.edenspiekermann.com



16

Lanzando la app

El momento estelar de una aplicación llega cuando finalmente es publicada en la tienda y para lograrlo, hay que cumplir una serie de pasos y requisitos. Después del lanzamiento, la historia continúa, escuchando a los usuarios y mejorando la app.

PUBLICACIÓN DE LA APP EN LAS TIENDAS OFICIALES

Una aplicación se publica después de la etapa de pruebas, cuando ya se tiene seguridad acerca de su correcto funcionamiento, estabilidad, desempeño y se considera libre de errores, tanto de usabilidad como de diseño.

El proceso de publicación en cada una de las tiendas es una tarea relativamente fácil y está bien documentada. Antes de comenzar, es importante prepararse y disponer de todos los recursos de diseño requeridos y hay que asegurarse de que la aplicación cumpla con las políticas de publicación de las tiendas para evitar que sea rechazada o bloqueada.

Una aplicación puede tardar varios días en ser aprobada, sobre todo cuando es la primera vez que se envía. Hay que tener

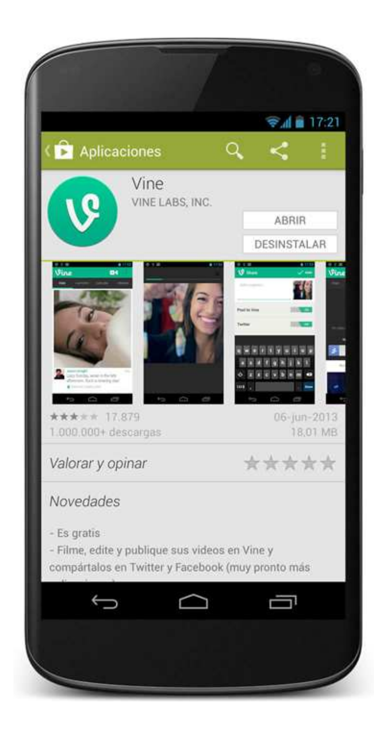


FIG 16.1. Íconos de lanzamiento y capturas de pantalla son esenciales para convencer al usuario de descargar la app.

en cuenta este tiempo, especialmente si se quiere hacer coincidir su salida al mercado con alguna fecha especial del año, como Navidad.

También cabe recordar que la publicación no es gratuita. Tanto en Google Play, como en App Store y en Windows Phone Store, hay que asumir un costo para iniciar el proceso que varía en cada uno de los casos.

En Google Play es necesario pagar 25 dólares por única vez al momento de crear la cuenta para la publicación. En el caso de App Store el pago es mayor, asciende a 99 dólares que, además, deberán pagarse anualmente; esta situación es idéntica a la que se presenta para desarrolladores de Windows Phone.

Crear la cuenta para publicación y realizar el pago correspondiente, permite acceder a una serie de herramientas de gestión y estadísticas sobre la cantidad de descargas de la app y su monetización.

Imágenes y elementos promocionales

Cuando se publica una aplicación, no solo debe subirse el archivo de la app, sino también aquellos elementos que la acompañarán en la página promocional de la tienda y que, en la mayoría de los casos, se trata de capturas de pantalla, textos descriptivos, ícono de lanzamiento e imágenes[1].

La importancia de preparar bien estos recursos radica en que serán visibles al público y servirán como herramientas de promoción y comunicación, lo cual puede ser determinante para la descarga de la aplicación. Textos e imágenes deben cuidar hasta el último detalle y cumplir con los requerimientos de cada una de las tiendas en cuanto a formato, dimensiones y resoluciones[2].

1 http://msdn.microsoft.com/en-us/library/windowsphone/help/jj206715(v=vs.105).aspx
2 https://support.google.com/googleplay/android-developer/answer/1078870

En el caso particular de las capturas de pantalla, es conveniente elegir las más representativas y atractivas de la aplicación. Si una pantalla no dice algo significativo de la app, la recomendación es no incluirla. También se pueden aprovechar estas imágenes para incluir, conjuntamente, otros gráficos que expliquen el funcionamiento de la app o refuercen el concepto que se quiere comunicar.

Cuando el contenido de las pantallas incluya nombres de personas o lugares, fotos o títulos, es aconsejable reemplazarlos por datos que simulen lo mejor posible una situación real, para que estas no luzcan como pantallas creadas durante los test de la aplicación.

Aunque pueda parecer un detalle menor, también es bueno tener en cuenta que la barra de estado superior del teléfono operativo debería estar lo más limpia posible de íconos que distraigan la atención del contenido principal —por ejemplo correos sin leer— aunque para ello haga falta retocar ligeramente la imagen y hacerla más presentable.

Se aconseja darle importancia a la elección de estos elementos ya que, junto al ícono de la app, son gráficos fundamentales para convencer al usuario: muchos de ellos tomarán la decisión de descargarla por lo que vean aquí.

El proceso de aprobación

Subir una aplicación no garantiza que esta sea publicada, ya que debe someterse a un proceso de aprobación que es diferente en cada tienda.

El proceso en Google Play es más abierto y tolerante[3] con una ventaja evidente: la mayoría de las aplicaciones son publicadas, a menos que incumplan claramente alguna de las políticas de la tienda, caso en el cual pueden ser retiradas y en última

3 https://support.google.com/googleplay/android-developer/?hl=en&rd=1#topic=2364761

instancia, si la situación lo requiere, puede ser suspendida la cuenta del desarrollador. Por supuesto, esta facilidad para publicar significa que pueden encontrarse muchas aplicaciones de dudosa utilidad entre todas las alternativas que ofrece la tienda.

Por el contrario, en App Store la aprobación es especialmente rigurosa, con una política pensada para garantizar cierto estándar de calidad en la publicación de aplicaciones, pero que en algunos casos puede rozar la exageración.

La guía completa de causas de rechazo está disponible al entrar a la cuenta de desarrollador. Solo por nombrar algunas, podemos mencionar aquellas que hacen referencia al contenido no original o similares a otras existentes, aplicaciones que no estén terminadas —por ejemplo si están en forma de demostración o prueba— o que representen una ofensa para personas, religiones o incluyan contenido inapropiado para niños.

En el caso de Windows Phone, la aplicación también debe pasar un proceso de certificación en el cual se verifica que cumpla las normas de la tienda. Los aspectos más importantes a tener en cuenta para que una app sea aprobada son la apariencia general —que no deje lugar a dudas sobre a qué sistema operativo pertenece, no contenga adornos ni elementos innecesarios y haga buen uso de los botones de sistema— y la calidad del contenido[4].

Publicación o distribución fuera de las tiendas

La opción de distribuir las aplicaciones a través de las tiendas oficiales es, sin duda, la más recomendable, ya que este sistema está apoyado en un mecanismo que se ocupa, entre otras cosas, de proveer los recursos necesarios para gestionar los pagos por

4 App certification requirements for Windows Phone. http://msdn.microsoft.com/en-us/library/windowsphone/develop/hh184843(v=vs.105).aspx

descarga y dentro de la aplicación, así como también la publicidad. Sin embargo, no es el único camino.

Las alternativas a Google Play

Android tiene varias opciones si se quiere evitar —o complementar— la publicación en Google Play, ya que esta no es la única forma. Existen otras tiendas que también ofrecen descargas de aplicaciones para este sistema operativo, como la tienda de apps de Amazon[5] o Samsung[6].

Alternativamente, se puede proponer la descarga directamente desde un sitio web. Con un enlace al archivo de la aplicación, cualquier persona puede descargar e instalar una app en su teléfono sin problemas. El único requisito es que el usuario debe tener habilitada en la configuración del móvil la opción de «Instalar aplicaciones de terceros»[7].

Otra forma de distribuir una aplicación en Android es a través del correo electrónico. Por tratarse únicamente de un archivo, puede enviarse fácilmente por este medio para que las personas la instalen. Si estas tienen la configuración lista para instalar aplicaciones, el botón «Instalar ahora» aparecerá en el mismo correo.

Esta opción es recomendable para usuarios de prueba y de confianza, y cuando se quiera mantener cierta independencia de las tiendas, con las limitaciones que ello implica. Claramente, el mayor inconveniente que presenta es que deja vía libre a la piratería y distribución no autorizada, ya que cualquier persona que cuente con el archivo puede enviarlo a un tercero, sin contar con el consentimiento para hacerlo.

5 http://www.amazon.com/mobile-apps/b?ie=UTF8&node=2350149011
6 http://apps.samsung.com/
7 http://developer.android.com/tools/publishing/publishing_overview.html

iOS y Windows Phone, solo para pruebas

En el caso de iOS y Windows Phone[8] la distribución independiente no es tan libre y masiva. Antes de ser incluida en App Store o en Windows Phone Store, la aplicación puede enviarse a usuarios individuales, pero previamente deben ser autorizadas, una por una, las cuentas de las personas que vayan a usar la app en su teléfono.

Esta forma de distribuir la aplicación no tiene otro fin más que ponerla a circular dentro de una empresa o hacerla llegar a un grupo selecto y limitado de usuarios, que pueda probarla y dejar comentarios antes de que la aplicación salga públicamente al mercado. Esta práctica es recomendable especialmente para garantizar un buen funcionamiento en diferentes terminales y versiones de sistemas operativos.

Después del lanzamiento

El trabajo en una aplicación no termina una vez que esta ha sido lanzada y está publicada en las tiendas. De hecho, ahí comienza una etapa más emocionante porque el producto está en manos de usuarios reales. La app se ha puesto su traje y finalmente está en la calle[9].

Esto significa que quienes la usen empezarán a compartir su experiencia e impresiones que, junto con las estadísticas de uso y descargas[10], servirán como referencia para mejorar la aplicación y corregir aquellas fallas de diseño o funcionalidad que, a pesar de los esfuerzos anteriores, se pasaron por alto.

8 Company app distribution for Windows Phone. http://msdn.microsoft.com/en-us/library/windowsphone/develop/jj206943(v=vs.105).aspx

9 Improving App Quality After Launch. http://developer.android.com/intl/es/distribute/googleplay/strategies/app-quality.html

10 http://appfigures.com/

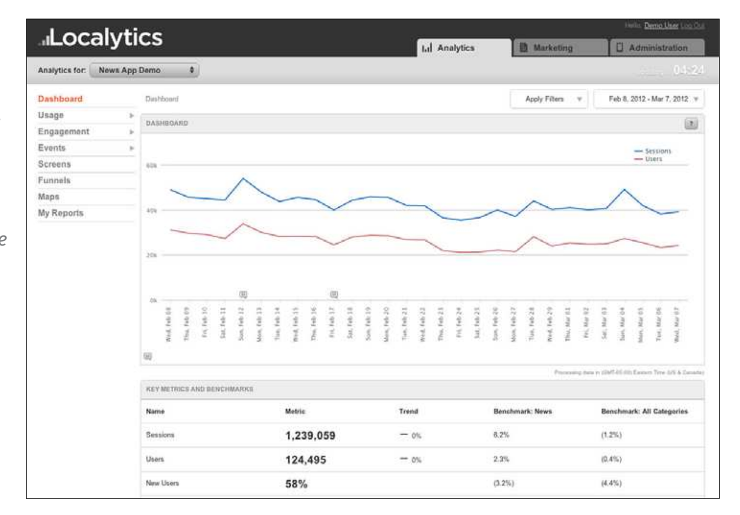


Fig 16.2. *Usar analíticas permite obtener información sobre el comportamiento de los usuarios y el desempeño de la app.*

Este tipo de mejoras conducirá a aumentar la calidad de la aplicación, lo que a su vez, se traducirá en un mayor número de descargas y más comentarios positivos. Los usuarios son quienes corren la voz de la aplicación y es mejor mantenerlos contentos. De la misma forma, las opiniones de la prensa y medios especializados pueden derivar en más descargas.

Comentarios de los usuarios

Una buena forma de retroalimentación son las opiniones de los usuarios. Y esto no se refiere solamente a los comentarios —casi siempre positivos— del círculo de amigos, sino a aquellos que provienen de personas que no tienen nada que ver con nuestro entorno.

Una vez subida a la tienda, cada aplicación recibe valoraciones y reseñas. Aunque no todas son de gran ayuda, es importante prestar atención a estos comentarios para identificar entre ellos los que sean de verdadera utilidad.

Nadie mejor que los usuarios que ya han descargado la aplicación para —queriendo o no— hacer pruebas con ella. Reportes

de fallos funcionales, opiniones sobre diseño o usabilidad, entre otras cosas, pueden encontrarse entre los comentarios de usuarios; seguirlos, es una buena herramienta para mejorar constantemente la app.

Analíticas de uso

Tal como suele hacerse en la web, también es posible recopilar datos estadísticos sobre el uso de la aplicación, a través de herramientas analíticas pensadas especialmente para móviles.

Las analíticas permiten estudiar el comportamiento de los usuarios al usar la aplicación, como patrones de navegación e interacción, entre otros. De esta forma, puede saberse, por ejemplo, cuáles fueron las pantallas más visitadas y cómo llegaron a ellas. También es posible conocer con cuáles botones y otros elementos interactúan más los usuarios. Otra información que puede ser de utilidad y se obtiene por medio de analíticas es el tiempo de retención, es decir, la cantidad de tiempo que pasan los usuarios dentro de la app.

Sin embargo, de toda la información que puede obtenerse sobre el uso de la aplicación, es importante centrar las analíticas en aquellos indicadores clave —*Key Performance Indicator* o KPI— que tengan sentido para el tipo de app que se ha desarrollado. Como vimos anteriormente, existen diferentes clases de aplicaciones y cada una de ellas requiere métricas distintas. Por ejemplo, para un juego puede ser importante estudiar el tiempo de retención y la cantidad de veces que el usuario vuelve a abrirlo. Pero en el caso de una red social, sería más interesante analizar cuántos contenidos nuevos se publican o cuántas veces son compartidos.

Obtener la información correcta conlleva a hacer un uso más eficiente de los datos. Un estudio a conciencia de las analíticas, se traducirá en una serie de mejoras que podrán estar

disponibles en futuras versiones o actualizaciones de la app, refinándola y mejorándola cada vez más.

Una vez que se tiene el panorama más o menos claro, puede empezarse a usar analíticas de forma inmediata, incluso, mucho antes de que la aplicación salga al mercado. Esto es especialmente útil cuando existen versiones alfa o beta de la app que se distribuyen antes de estar oficialmente disponible en las tiendas. Igualmente, integrar las analíticas con anticipación es una buena forma de enterarse si se han instalado correctamente, verificando que los primeros datos obtenidos tengan el aspecto deseado. Es altamente recomendable que en el momento de hacerla pública, la aplicación ya tenga las analíticas preparadas para obtener información desde el minuto cero.

En el mercado existen diferentes herramientas que permiten gestionar analíticas. Algunos ejemplos son Google Analytics[11], Mixpanel[12], Flurry[13] o Localytics[14]. Con características diferentes, elegir una u otra, dependerá de ciertos parámetros como la información que permiten obtener, su costo o la cantidad de usuarios que tendrá la aplicación.

Promoción

Para conseguir el mayor número de descargas posibles es necesario promocionar la aplicación, para que los potenciales usuarios se enteren de su existencia y la conozcan. Las diferentes tiendas se ocupan en parte de esta tarea; sin embargo, no se les puede dejar todo el trabajo de promoción a ellas o a los buscadores que deben encontrar la app. Existen otras formas de hacerlo por cuenta propia: a través de campañas de marketing, anuncios en Internet o reseñas de la app en blogs especializados.

11 http://www.google.com/analytics/features/mobile.html
12 https://mixpanel.com/
13 http://www.flurry.com/
14 http://www.localytics.com/

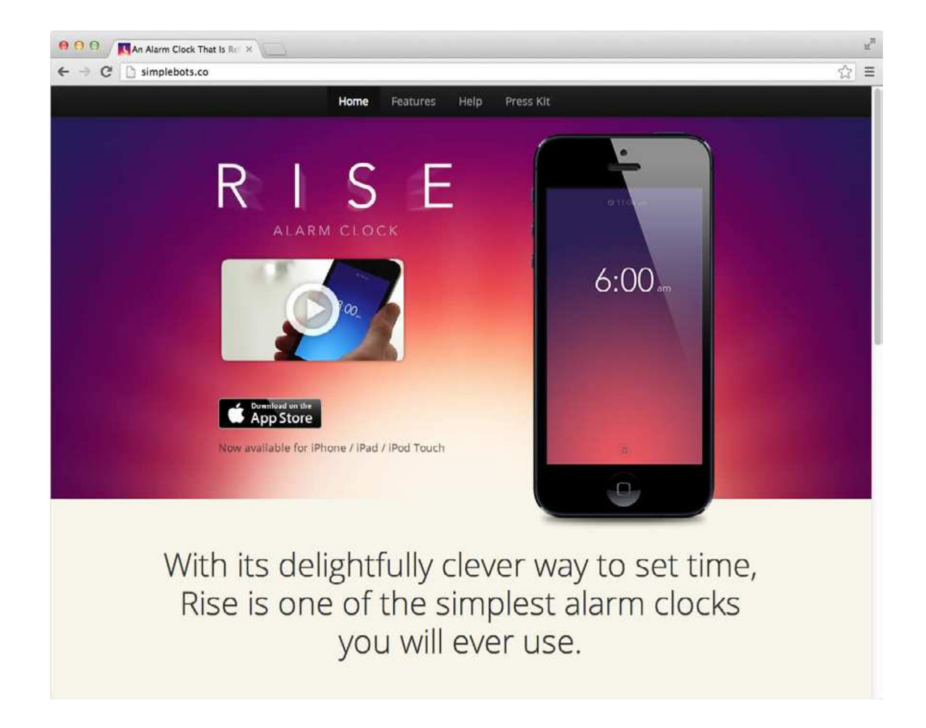


FIG 16.3. Rise, muestra la información esencial y un enlace a la tienda con un tratamiento visual atractivo.

Una forma muy eficaz de promocionar una aplicación es por medio de un sitio web que, contrario a las webs corporativas o de otro tipo de productos, no tiene porqué ser demasiado complejo ni contar con muchos niveles de navegación.

Es indispensable contar con una estructura mínima pero bien trabajada. La landing page es la página donde «aterrizarán» aquellos internautas que hayan dado con la dirección web de la aplicación. En esta página es necesario responder algunas preguntas iniciales que pueda tener el visitante ocasional: ¿Qué hace la aplicación? ¿Es gratis? ¿Para qué teléfonos está disponible? La gráfica y el mensaje utilizados deben ayudar a despejar estas dudas en el menor tiempo posible.

En el caso de Android, que permite la descarga independiente de la tienda, otra función del sitio web es servir como

canal de distribución, proveyendo enlaces para obtener la app directamente.

No obstante, la mejor opción es usar la página inicial como puente para llegar a la tienda donde se encuentra la aplicación. Cada una de las tiendas ofrece diferentes elementos promocionales con su identidad para incluir en la web, como botones de descarga entre otros[15].

Las actualizaciones

Las observaciones que se obtienen a través de los comentarios y estadísticas de uso, se transforman en una serie de mejoras que pueden implementarse en la aplicación. Es entonces cuando deja de existir solo la versión inicial y se tiene una app mejorada, lista para sustituir a la anterior. Esto es posible a través de las actualizaciones[16].

Cuando lo que se ha corregido en la app son fallos realmente graves, que impiden el uso futuro de la aplicación, puede realizarse lo que conocemos como «actualización forzosa». Como el nombre lo indica, es aquella que obliga a los usuarios a descargar la última versión. El riesgo de este tipo de actualizaciones es que puede hacer que algunos usuarios dejen de usar la aplicación porque no quieren actualizarla y tampoco pueden usar la versión que tienen instalada. En otros casos, la descarga de las actualizaciones suele ser opcional y puede realizarse cuando el usuario lo prefiera.

Las actualizaciones generalmente responden a un ciclo de desarrollo, que dura una determinada cantidad de días, al cabo del cual se liberan. De esta forma, se va trabajando en la prueba e implementación de cambios futuros que resultarán

15 http://developer.android.com/intl/es/distribute/googleplay/promote/badges.html

16 Warren, Christina. When & How You Should Update Your Mobile App. http://mashable.com/2011/09/22/mobile-app-update/

en una nueva versión de la aplicación. La ventaja de esta forma de trabajar es que se realiza una mejora constante de la app y los usuarios saben que pueden esperar una versión mejorada cada cierto tiempo.

En Android y iOS las aplicaciones se actualizan automáticamente, evitando al usuario la preocupación de asegurarse tener las últimas versiones de las apps instaladas. Por el contrario, en Windows Phone sigue siento necesaria su intervención, ya que tiene que actualizar manualmente las apps, cuando el SO así se lo indica con una notificación visual.

Es importante recalcar que subir una nueva versión de la aplicación a la tienda de Apple o de Microsoft obliga a someterla nuevamente a la revisión del cumplimiento de las políticas por parte de estas tiendas. En el caso de Google Play, como hemos comentado, el proceso es más simple y directo.

17

Firefox OS y Ubuntu para teléfonos: Los nuevos jugadores

Si bien a lo largo del libro nos hemos concentrado solamente en Android, iOS y Windows Phone, la realidad es que existen más competidores entre los sistemas operativos para móviles[1].

Tal es el caso de Firefox OS[2] y Ubuntu[3] para teléfonos, tan solo dos de los representantes de un nuevo grupo de jugadores —que también incluye a Tizen y Sailfish— que quiere abrirse camino entre los gigantes. Casualmente, ambos tienen una filosofía de código abierto y apuestan por el HTML como base de su estructura.

Lo anterior quizás pueda ser un cambio de paradigma respecto a los grandes del mercado y una nueva manera de concebir las aplicaciones para móviles, que afecta tanto el diseño como el desarrollo y la interacción con los usuarios.

Entrevistamos a los responsables de diseño de Firefox OS y Ubuntu, buscando entender en qué se distinguen estos sistemas operativos y qué podemos esperar de ellos, en cuanto a diseño se refiere.

Firefox OS

El sistema operativo de la fundación Mozilla ya se encuentra en algunos teléfonos del mercado y promete llegar a muchos más en los próximos meses. Es la primera apuesta de los creadores del famoso navegador por el mundo de los móviles.

Patryk Adamczyk es el Jefe de Diseño de Firefox OS y nos cuenta qué es lo que separa a Firefox OS de otros sistemas operativos para móviles:

> *El objetivo de Mozilla es mantener una web abierta y permitir a todo el mundo tener acceso a una experiencia*

1 http://en.wikipedia.org/wiki/Mobile_operating_system
2 http://www.mozilla.org/es-ES/firefox/partners/
3 http://www.ubuntu.com/phone

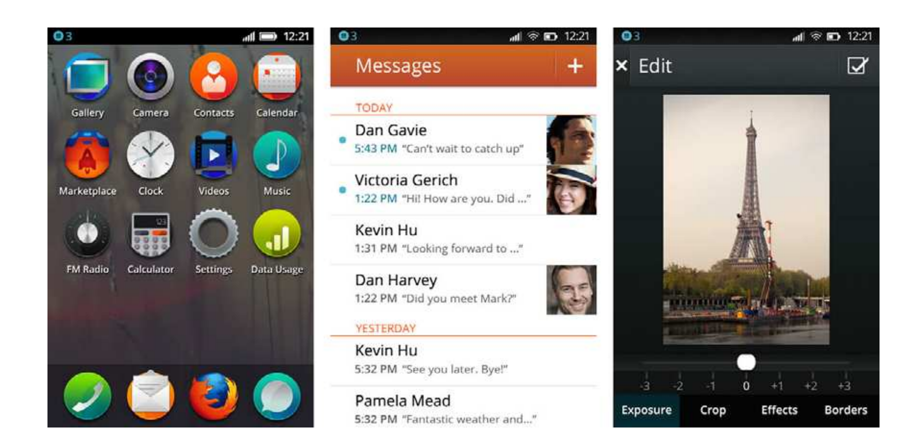


FIG 17.1. *Firefox OS intenta encontrar su identidad visual, al mismo tiempo que presenta claras influencias de otros SO.*

completa. Firefox OS no solo viene con un navegador completamente funcional sino que también está construido usando tecnologías web, permitiendo a millones de desarrolladores escribir aplicaciones para la plataforma, sin necesidad de aprender un nuevo lenguaje.

De muchas formas, Firefox OS funciona como sus competidores, ya que se posiciona para usuarios que están cambiando sus smartphones desactualizados o feature phones y aspira a hacer esa transición fluida. En nuestra versión 1, el costo del dispositivo y la curva de aprendizaje son bajos; esperamos alcanzar los mercados emergentes donde el acceso a Internet está sin explotar y la mayoría de personas accede a través de sus ordenadores hogareños, locutorios o cafés.

Hemos diseñado la mayoría de los componentes usando CSS para conseguir velocidad y escalamiento. Mientras vamos avanzando en actualizaciones graduales durante los próximos meses, nuestro plan es evolucionar la interfaz para que se sienta aún más como una web.

Los principios de diseño de Firefox OS[4] fueron la base sobre la que se construyó su estilo visual[5]. Aunque podemos esperar que la apariencia de la interfaz cambie y evolucione con el tiempo, por lo visto hasta ahora, Firefox OS es visualmente más cercano a Android, pero también con algo de influencia de iOS y Windows Phone.

En el diseño de este SO también se intenta privilegiar el contenido sobre el contenedor, dándole mayor importancia al contenido visual —como videos o fotografías— que se apoya sobre una retícula variable para cada necesidad, complementada con fondos oscuros para destacar los gráficos.

El estilo visual de Firefox OS cuenta con texturas y volúmenes —aunque de una forma bastante medida y controlada—, sobre todo en aplicaciones como el reloj que da la apariencia de estar construido con materiales reales.

Para Firefox, la tipografía siempre ha sido importante y esto también se ve reflejado en su sistema operativo para móviles. La tipografía Meta, de Erik Spiekermann, utilizada tradicionalmente como sello de identidad de la marca, ha evolucionado en una versión para móviles llamada Fira, creada por el mismo diseñador.

Sin embargo, tal vez uno de los detalles más distintivos de Firefox OS, son los íconos circulares de la pantalla de inicio, algo completamente diferente a los otros sistemas operativos.

Habrá que ver cómo crece y evoluciona Firefox OS en los meses por venir, para saber si realmente consigue establecer una identidad visual que lo distinga de los demás.

4 Adamczyk, Patryk. MozCamp Warsaw: Design Principles Behind Firefox OS UX. https://blog.mozilla.org/ux/2012/09/mozcamp-warsaw-design-principles-behind-firefox-os-ux/

5 http://www.mozilla.org/en-US/styleguide/products/firefox-os/



Ubuntu para teléfonos

Fig 17.2. El desembarco de Ubuntu en teléfonos promete ser una buena alternativa a los SO actuales.

Ubuntu es una distribución de Linux conocida desde hace tiempo. Desarrollada por Canonical, es la preferida de muchos «linuxeros» y hasta el día de hoy había concentrado sus esfuerzos en el sistema operativo para ordenadores.

No obstante, las cosas empezaron a cambiar cuando hizo sus primeras incursiones en Android y de allí, como consecuencia lógica, comenzó a desarrollar su propuesta para móviles, que no es otra cosa que una extensión de su sistema operativo.

Ivo Weevers, Jefe de Diseño de Ubuntu, es la persona más indicada para contestar qué hace a Ubuntu diferente a los demás sistemas operativos para teléfonos:

> *Ubuntu es la única plataforma que ofrece una aproximación de diseño verdaderamente convergente. El mismo núcleo del sistema operativo se ejecuta en todos los formatos, incluyendo teléfonos, tabletas, ordenadores de escritorio y televisores inteligentes.*

La interfaz de usuario se escala perfectamente para adaptarse a los diferentes tamaños de pantalla y métodos de entrada. Esto permite soluciones únicas, tales como ejecutar una versión de escritorio desde un teléfono o construir una app que podrá ejecutarse en cualquier lado.

Sin duda, para un sistema operativo tan ubicuo es un desafío mantener unos lineamientos de diseño que se respeten en cada uno de los soportes.

¿Cuáles serán, entonces, los principios de diseño de Ubuntu para móviles[6]?

Ubuntu usa gestos deslizantes simples y naturales desde los bordes de la pantalla, para facilitar el acceso al contenido y cambiar entre aplicaciones.

Cada borde tiene un propósito específico —lo cual lleva la aproximación de borde / gesto más lejos que cualquier otra solución—, haciendo accesibles instantáneamente todas las aplicaciones, contenidos y controles, sin necesidad de navegar de nuevo por la pantalla inicial.

Las pantallas táctiles permiten teclados contextuales que solo aparecen cuando el usuario lo necesita. Ubuntu lleva este concepto un paso más allá, mostrando también controles de la interfaz solo cuando el usuario lo requiere.

Hemos usados tres principios clave en el diseño:

1. *Foco en el contenido.*
2. *Interacciones rápidas y naturales.*
3. *Estilo sofisticado.*

6 http://design.ubuntu.com/apps/get-started/design-vision

> *Aunque algunos de estos estilos de diseño son también mencionados por otras plataformas, Ubuntu es la única que realmente lo consigue a través del modelo de interacción de los bordes.*

Desafortunadamente, este sistema operativo sólo está disponible —de momento— para móviles Nexus. Quienes tengan otros modelos, tendrán que esperar un poco más para conocer su propuesta.

La expectativa es grande por saber cómo se transformará el mercado y cómo se desarrollarán simultáneamente estas opciones de software, para comprobar si —de una forma u otra— empiezan a influir en las demás.

18

Reflexiones finales

En los capítulos de este libro hemos visto cómo diseñar y desarrollar una aplicación, desde la idea inicial hasta su publicación, cuando se encuentra disponible para todo el mundo.

En cada etapa hemos desarrollado los conceptos que consideramos más interesantes y a la vez, aquellos que no necesariamente se encuentran en los manuales ni en las guías de cada sistema operativo. Estos últimos, son el fruto de nuestra experiencia, de lo que hemos aprendido durante estos años de trabajo. Sin duda, es lo que hubiéramos querido que alguien nos explicara antes de comenzar.

Esperamos que saques el máximo provecho de nuestras palabras. Ante todo, tómalas como un punto de partida desde el cual puedas investigar más. Cada uno de los temas tiene un riqueza infinita, imposible de abarcar en un solo libro; por ejemplo, existen publicaciones dedicadas a hablar solamente de test de usabilidad. Seguramente te habrás interesado por algún tema más que por otro, por lo cual te recomendamos usar los enlaces allí dispuestos para ampliar tus conocimientos.

Lo mejor empieza ahora

Una vez hayas terminado este libro, estás listo para poner en práctica sus contenidos. No hace falta esperar que algún cliente te pida un proyecto nuevo, puedes comenzar a hacer tus propias aplicaciones; aunque no se transformen inmediatamente en productos publicados y rentables, te servirán para investigar y empezar a plantear tu propia manera de resolver las cosas.

Incluso, solo con un teléfono puede ser suficiente para iniciarte en este mundo, usándolo a conciencia y observando crítica y detalladamente cómo hacen las aplicaciones los demás. Esta forma de interactuar con tu móvil debería convertirse, desde hoy, en una práctica habitual.

Aprovecha herramientas básicas como el lápiz y el papel para trasladar tus ideas, volcar conceptos y evolucionarlos. Para

hacerlo, no necesitas prácticamente nada más que tu imaginación. Quién sabe, quizás con el tiempo, esos esquemas rápidos se transformen en una app exitosa.

El papel del diseñador

Definitivamente, creemos en un diseñador que va más allá del estereotipo tradicional de constructor de interfaces. El diseño es una disciplina realmente interesante, llena de posibilidades y caminos que se bifurcan, especialmente a partir de su relación con la tecnología. Por ejemplo, en una aplicación podemos hablar tanto de diseño de información, como de interacción o visual.

Es normal sentirse más identificado, o incluso apegado, con determinados roles en un proyecto. Ejecutar el mismo papel de forma continua, con el tiempo, nos convierte en especialistas; pero, lo que realmente nos hace mejores profesionales, es tener un conocimiento que trascienda las fronteras de nuestro campo de acción, en este caso, saber cómo se idean y ejecutan otras etapas del diseño y desarrollo de una app. De esta forma, la relación e integración con otros miembros del equipo puede ser más fructífera, al generar sinergia que resulte en aplicaciones de mayor calidad.

Un diseñador debería entonces, en mayor o menor medida, involucrarse en todas las etapas del proceso, incluso en aquellas que le resulten más técnicas o tediosas. Esto no es algo que deba evitarse, por el contrario, debería aprovecharse para trasladar el aprendizaje al trabajo propio.

De la misma manera que los diseñadores deben dar un paso más allá, también los desarrolladores deberían entender el trabajo de diseño y ser conscientes del impacto que tiene en los usuarios una interfaz que, además de verse bien, cuida aspectos de usabilidad.

Esta relación puede permitir al programador incorporar una visión estética y de sentido visual, que le ayude a tomar decisiones a la hora de construir una aplicación, guiado también por su instinto y sin depender enteramente del diseñador.

Ya no decimos «los profesionales del futuro». Los profesionales de hoy más apreciados por pequeñas y medianas empresas son aquellos que pueden resolver y plantear soluciones que van más allá de sus tareas. Te invitamos a que seas uno de ellos.

El futuro de las apps

El diseño de aplicaciones está en un momento ideal para sumergirse de lleno en él. Si bien cada sistema operativo tiene su propuesta, suelen ser realmente cambiantes y abiertos a las innovaciones que proponen los diseñadores. Los patrones no son completamente cerrados, lo cual también deja lugar a la iniciativa del diseñador —ya nos comentó esta situación alguien tan reconocido como Loren Brichter en la entrevista—.

Al mismo tiempo, nuevos sistemas operativos para móviles, como Firefox OS y Ubuntu, están emergiendo, por lo que podemos esperar que en los próximos meses el panorama general siga cambiando. Desde nuestro lugar, podemos intuir que la filosofía de diseño de cada plataforma se verá afectada por las demás, tal como en su momento el estilo plano de Windows Phone sentó un precedente que aprovecharon sus competidores.

Son días emocionantes y, seguramente, lo mejor está por venir. Prepárate para ello.

Involúcrate

Seguramente ahora que has terminado de leer el libro, tienes ganas de empezar a diseñar tu app, pero puede que también tengas dudas e inquietudes en tu cabeza.

Antes de dejarte ir, queremos animarte a que si tienes preguntas, comentarios o sugerencias sobre cualquier tema del libro, puedes hacérnoslas llegar. Estaremos encantados de escucharte por correo (**hola@appdesignbook.com**), o por las cuentas de Twitter de «Simón» (**@millonestarde**) o José (**@josevittone**).

Las ventajas de escribir un libro digital es que puede actualizarse a través del tiempo, con relativa facilidad. Contenidos que quedan obsoletos, datos incorrectos o enlaces que no funcionan pueden ser mejorados en futuras versiones que te haremos llegar oportunamente, para que siempre tengas en tus manos la última versión de *Diseñando apps para móviles*.

Anímate también a visitar nuestra web (**appdesignbook.com**). Todos los contenidos del libro están disponibles ahí también, gratis para todo el mundo. Es una excelente oportunidad para empezar a interactuar con nosotros, y con otros lectores que —como tu— tienen ganas de llevarse el mundo de las apps por delante.

Te esperamos.

Javier «Simón» Cuello
José Vittone

Glosario

En las siguientes páginas encontrarás una explicación más clara y detallada de algunos términos usados en este libro. Vale aclarar que, aunque en ocasiones el significado de alguno de ellos puede trascender el contexto de las aplicaciones, hemos preferido acotar la descripción al ámbito de los contenidos.

Accesibilidad: Posibilidad de acceso a los contenidos por cualquier persona independientemente de sus capacidades físicas. A nivel visual está determinada, entre otras cosas, por el tamaño de los textos y botones y por el contraste que estos elementos tienen con el fondo. Una app accesible también hace referencia a una correcta programación del código que permite, por ejemplo, que los contenidos puedan ser interpretados por accesorios para ciegos.

App: Es el nombre usado comúnmente para referirse a las aplicaciones, que surge de acortar el vocablo inglés application. Es una pieza de software que se ejecuta en teléfonos móviles y tabletas y, como te habrás dado cuenta, es el objeto de estudio de este libro. Si aún no entiendes lo que es una app te recomendamos leer este libro con más atención.

Benchmarking: El *benchmarking* es un proceso sistemático para evaluar comparativamente productos, servicios y procesos. En nuestro contexto, es entonces el estudio comparativo y analítico de otras aplicaciones, con el fin de determinar la calidad y características de cada una de ellas, tomándolas como parámetros de referencia.

Compilar: Es la acción de empaquetar un código. El resultado de compilar el código de una aplicación es el archivo final que está listo para ser subido a la tienda.

Contexto de uso: Entorno general conformado por la ubicación y espacio físico que rodea al usuario y al dispositivo. El contexto de uso determina, además, la forma en que estos dos componentes se relacionan e interatúan entre sí.

CSS: Siglas de *Cascading Style Sheets*, que en español sería «Hojas de estilo en cascada». Ya sea en archivos separados o dentro del código HTML, este lenguaje determina la apariencia visual de una web o aplicación web definiendo, entre otras cosas, los colores y tamaños de fuente.

Densidad de pantalla: Es la cantidad de píxeles por espacio físico que tiene una pantalla. Generalmente se mide en «píxeles por pulgada» o DPI por las siglas en inglés de *Dots per inch*. Las densidades son diferentes por cada modelo de móvil y se dividen por lo general en bajas, medias o altas, denominación que puede variar dependiendo del sistema operativo.

DP: Corresponde a las siglas de *Density-independent pixels* o «píxeles independientes de la densidad». Es una unidad de medida empleada por Android que está relacionada con la densidad física de la pantalla. Los DP son unidades relativas a las pantallas de 160 DPI, en las cuales un DP equivale a un píxel.

El uso de DP en lugar de píxeles es una solución propuesta para adaptar correctamente una interfaz cuando se lleva a otras densidades de pantalla.

Escenario: Se refiere a la combinación de contexto de uso y «Persona». Determina la manera como un usuario se relaciona con el móvil en una situación específica.

Experiencia de usuario o UX: Concentra las emociones y percepciones que tiene una persona al usar una interfaz o producto. En el caso de las apps, está influida por un conjunto de factores que determinan si la experiencia es positiva o negativa, entre ellos, la accesibilidad, diseño visual, diseño de interacción y usabilidad.

Feedback: Es la respuesta, generalmente inmediata, de la interfaz para mantener al usuario informado de las acciones que acaba de realizar. En este sentido, puede ser la confirmación de éxito o de error obtenida al ejecutar una tarea y puede manifestarse a través de avisos o por medio de elementos visuales más sutiles.

El *feedback* también puede referirse a las observaciones y comentarios de usuarios, que sirven como parámetros o indicadores para mejorar una app.

HTML: Corresponde a las siglas de *HyperText Markup Language*. Es el lenguaje que se utiliza tradicionalmente para construir páginas web y aplicaciones web para móviles. Define la estructura de un documento web basado en una serie de etiquetas.

Interfaz o UI: La interfaz o *User Interface* es la capa que existe entre el usuario y el dispositivo, que le permite interactuar con este último. En las aplicaciones se trata del componente gráfico que contiene elementos que producen reacciones al pulsarlos y permiten al usuario realizar tareas, como también, aquellos estáticos sobre los cuales se interpretan los contenidos.

JavaScript: Lenguaje de programación utilizado principalmente en proyectos web como sitios o aplicaciones, que muchas veces actúa en conjunto con HTML y CSS para dotarlos de funcionalidad.

KPI: Del inglés *Key Performance Indicator*, son los «indicadores clave de desempeño» que miden las variables de ejecución de un proceso, con el fin de obtener datos relevantes para determinar el rendimiento general y conocer si se han alcanzado los objetivos fijados.

Librería: En programación, se llama así al conjunto de código externo que puede aprovecharse para conseguir determinados comportamientos. Tiene relación directa con el lenguaje de programación elegido.

Monetizar: Acción de obtener rédito económico. En una aplicación, tiene que ver con las formas de obtener ingresos de ella y está directamente relacionada con el modelo de negocio y la estrategia comercial.

Móvil: También llamado (teléfono) celular en algunos países de América Latina, es un artefacto electrónico de tamaño variable donde funcionan las aplicaciones y estamos casi seguros de que tienes uno en tu mano o bolsillo ahora mismo.

Orientación: Es la manera en que se muestra el contenido en pantalla, dependiendo de la forma en que el usuario sostiene su tableta o teléfono. Puede ser vertical u horizontal.

Persona: Personificación de los usuarios que surge como resultado de estudios basados en su comportamiento y características etnográficas. Esta investigación está basada en los patrones comunes que se detectan en los usuarios. El concepto de «Persona» fue creado por Cooper y es una herramienta habitual en el diseño de interacción.

Píxel o px: Unidad física mínima formada por puntos de color que se reparten a lo largo de la superficie de una pantalla. En diseño también suele usarse como unidad de medida para los componentes gráficos de la interfaz en los diferentes programas de edición.

Ranking: Clasificación ordenada que se da a las aplicaciones en cada una de las tiendas, dependiendo de factores como cantidad de descargas o cantidad de valoraciones positivas.

Resolución de pantalla: Es la cantidad de píxeles que puede ser mostrada en la pantalla de un dispositivo y consiste en una relación entre el ancho y alto de la misma.

SDK: El *Software Development Kit* o «Kit de desarrollo de software» provee a los programadores herramientas necesarias para desarrollar el código de una aplicación. Tanto Android, como iOS y Windows Phone, ofrecen uno diferente.

Simulador: Un simulador permite probar la aplicación sin necesidad de contar con un móvil. De esta forma, se puede ejecutar el código en el ordenador y ver los resultados en la pantalla, con el fin de realizar comprobaciones preliminares sobre el funcionamiento de la app.

Sistema operativo o SO: Es el software que contiene cada uno de los teléfonos y sobre el cual se ejecutan las aplicaciones. Las distintas versiones de Android, iOS y Windows Phone, son ejemplos de sistemas operativos.

Sp: En Android se llama así a los *Scale-independent pixels* o «píxeles independientes de la escala», usados para textos. En diseño, estos tienen el mismo comportamiento que los DP, con la diferencia que el tamaño medido en sp también puede ser afectado por las preferencias del usuario.

Tamaño de pantalla: Es el tamaño físico que tiene la pantalla de un extremo al otro, en forma diagonal y medido, generalmente, en pulgadas.

Tema: Combinación de colores que usan Android y Windows Phone de forma preestablecida. En Windows Phone, el usuario puede elegir entre una serie de temas que afectan el color de fondo y elementos destacados a través de las pantallas de todo el sistema operativo.

Tienda: Es el canal de distribución y comercialización de aplicaciones desde donde pueden descargarse de forma gratuita o paga. Cada uno de los sistemas operativos móviles mencionados en este libro tiene una tienda oficial; sin embargo, en el caso de Android, existen varias opciones alternativas además de Google Play como las tiendas de apps de Amazon o Samsung.

Usabilidad: En su sentido más amplio, está relacionada con la eficacia y eficiencia de la interfaz de una aplicación para permitir a un usuario determinado realizar una tarea o cumplir un objetivo. La usabilidad no puede analizarse de forma aislada, ya que está vinculada con un contexto particular y un usuario específico; por tanto, está directamente asociada a la experiencia de usuario.

Usuario: El usuario es quien realiza interacciones con la aplicación a través de su interfaz. Es el foco del llamado «diseño centrado en el usuario» que tiene como eje sus necesidades, para proponer soluciones que resuelvan los problemas, considerando sus emociones y expectativas.

Créditos de imágenes

Presentación

Foto de las ilustraciones por PÉREZ, Catalina. enseguida.net/collections/tuga-prints/

1. Las aplicaciones

1.1. Elaboración propia a partir de imagen de Apple. 2013.
1.2. WALTON, Trent. 2012. http://trentwalton.com/2012/10/03/a-new-microsoft-com.
1.3. Elaboración propia. 2013.
1.4. Elaboración propia. 2013.
1.5. Elaboración propia a partir de capturas de pantalla de Facebook. 2013.
1.6. Elaboración propia a partir de capturas de pantalla de Netflix. 2013.

2. Entendiendo las posibilidades

2.1. Elaboración propia a partir de captura de pantalla de Angry Birds. 2013.
2.2. Elaboración propia a partir de captura de pantalla de Path. 2013.
2.3. Realmac Software. 2013. Recuperada de: https://itunes.apple.com/es/app/clear/id493136154?mt=8.
2.4. Sophiestication Software. Kit de prensa. 2013.
2.5. Elaboración propia a partir de captura de pantalla de Paper. 2013.
2.6. Elaboración propia a partir de capturas de pantalla de Cute the Rope y Whatsapp. 2013.
2.7. Elaboración propia a partir de captura de pantalla de Line. 2013.
2.8. Halfbrick. 2013.

2.9. Elaboración propia. 2013.

3. Irene Pereyra: UX con magia en Fi

3.1. Fi. 2013. Recuperada de: http://www.f-i.com/nickelodeon/kids-choice-awards.

4. Explorando ideas

4.1. Elaboración propia a partir de capturas de pantalla de iBulb, iLed FlashLight y Linterna. 2013.
4.2. De Sousa, Fernando. 2013.
4.3. Dropbox Inc. Kit de prensa. 2013.

5. Definiendo la propuesta

5.1. Elaboración propia. 2013.
5.2. Elaboración propia. 2013.
5.3. Elaboración propia a partir de captura de pantalla de Instagram. 2013.
5.4. Elaboración propia a partir de artículo en UX Matters. 2012. Recuperado de: http://www.uxmatters.com/mt/archives/2012/05/expressing-ux-concepts-visually.php.
5.5. Estudio Chipsa. 2013. www.chipsa.ru.
5.6. Stark, James.
5.7. Arnall, Timo.
5.8. Android. 2013. Recuperada de: http://developer.android.com/intl/es/design/downloads/index.html.
5.9. Elaboración propia a partir de captura de pantalla de Omnigraflle e imagen de 2013.

6. Dustin Barker: Electronic Banking Made Simple

6.1. Elaboración propia a partir de imágenes de Simple. Kit de prensa. 2013.
6.2. Elaboración propia a partir de imágenes de Simple. Kit de prensa. 2013.

7. Interacción y patrones

7.1. Elaboración propia a partir de imagen de HOOBER, Steven. 2013. Recuperada de: http://blog.utest.com/how-mobile-users-hold-devices/2013/03/
7.2. Elaboración propia a partir de captura de pantalla de Weightbot. 2013.

7.3. Elaboración propia. 2013.
7.4. Elaboración propia. 2013.
7.5. Elaboración propia. 2013.
7.6. Elaboración propia. 2013.
7.7. Elaboración propia. 2013.
7.8. Elaboración propia. 2013.
7.9. Elaboración propia. 2013.
7.10. Elaboración propia. 2013.
7.11. Elaboración propia. 2013.
7.12. Elaboración propia. 2013.
7.13. Elaboración propia. 2013.
7.14. Elaboración propia. 2013.
7.15. Elaboración propia. 2013.
7.16. Elaboración propia. 2013.
7.17. Elaboración propia. 2013.
7.18. Elaboración propia. 2013.
7.19. Elaboración propia a partir de captura de pantalla de Google. 2013.

8. Diseño visual

8.1. Elaboración propia a partir de captura de pantalla de Gmail. 2013.
8.2. Elaboración propia a partir de captura de pantalla de iOS. 2013.
8.3. Elaboración propia a partir de captura de pantalla de Windows Phone. 2013.
8.4. Elaboración propia a partir de captura de pantalla de Wunderlist. 2013.
8.5. Elaboración propia a partir de captura de pantalla de Vine. 2013.
8.6. Elaboración propia a partir de captura de pantalla de Google Play. 2013.
8.7. Elaboración propia. 2013.
8.8. Elaboración propia. 2013.
8.9. Elaboración propia. 2013.
8.10. Elaboración propia. 2013.
8.11. Elaboración propia. 2013.
8.12. Elaboración propia a partir de captura de pantalla de Skype, rdio y Nike+ Running. 2013.
8.13. Elaboración propia a partir de captura de pantalla de iOS. 2013.

8.14. Android. 2013. Recuperada de: http://developer.android.com/intl/es/design/style/metrics-grids.html.
8.15. Elaboración propia a partir de captura de pantalla de iOS. 2013.
8.16. Elaboración propia a partir de captura de pantalla de Windows Phone. 2013.
8.17. Elaboración propia. 2013.
8.18. LÜSCHER, Christoph. 2012. Recuperada de: http://ia.net/blog/responsive-typography-the-basics/
8.19. Android. 2013. Recuperada de: http://developer.android.com/intl/es/design/style/typography.html.
8.20. Elaboración propia a partir de captura de pantalla de Bloomberg. 2013.
8.21. Elaboración propia. 2013.
8.22. Elaboración propia. 2013.
8.23. Elaboración propia. 2013.
8.24. Elaboración propia. 2013.
8.25. Elaboración propia a partir de captura de pantalla de The Guardian. 2013.
8.26. Microsoft. 2013. Recuperada de: http://msdn.microsoft.com/en-us/library/windowsphone/design/hh202878(v=vs.105).aspx.
8.27. Android. 2013. Recuperada de: http://developer.android.com/intl/es/design/style/color.html.
8.28. Elaboración propia a partir de recursos de Teehan + Lax. 2013. Recuperados de: http://www.teehanlax.com/tools/iphone/.
8.29. Captura de pantalla de Letterpress. 2013.
8.30. Captura de pantalla de Gmail. 2013.
8.31. Captura de pantalla de Spotify. 2013.
8.32. Captura de pantalla de Google+. 2013.
8.33. Elaboración propia a partir de capturas de pantallas de Clear. 2013.
8.34. Elaboración propia a partir de captura de pantalla de Windows Phone. 2013.

9. Dustin Mierau: Path y el valor de los detalles

9.1. Elaboración propia a partir de captura de pantalla de Path. 2013.
9.2. Elaboración propia a partir de captura de pantalla de Path. 2013.

10. Probando con usuarios

10.1. Elaboración propia a partir de imagen de BULT, Mark. 2013.
10.2. Mr. Tappy. Kit de prensa. 2013.
10.3. MELBOURNE, Ben. 2013.

11. Preparando los archivos

11.1. OpenSignal. 2013. Recuperada de: http://opensignal.com/reports/fragmentation.php.
11.2. Android. 2013. http://developer.android.com/design/style/devices-displays.html.
11.3. YERMOLAYEV, Nikita. 2013. www.komarov.mobi.
11.4. Elaboración propia. 2013.
11.5. Elaboración propia. 2013.
11.6. LEMEDEN, Reda. 2013.
11.7. Elaboración propia. 2013.
11.8. Elaboración propia. 2013.
11.9. Elaboración propia. 2013.

12. Loren Brichter: Rompiendo todos los esquemas

12.1. laboración propia a partir de imagen de Atebis. Kit de prensa. 2013.
12.2. iMore. 2013. Recuperada de: http://www.imore.com/hall-fame-loren-brichter-and-tweetie.

13. Ejemplos a seguir

13.1. Captura de pantalla de 500px. 2013.
13.2. Captura de pantalla de 500px. 2013.
13.3. Captura de pantalla de Google Drive. 2013.
13.4. Captura de pantalla de Google Drive. 2013.
13.5. Captura de pantalla de The Guardian. 2013.
13.6. Captura de pantalla de The Guardian. 2013.
13.7. Captura de pantalla de National Parks. 2013.
13.8. Captura de pantalla de National Parks. 2013.
13.9. Captura de pantalla de Google Maps. 2013.
13.10. Captura de pantalla de Google Maps. 2013.
13.11. Captura de pantalla de Instapaper. 2013.
13.12. Captura de pantalla de Instapaper. 2013.
13.13. Captura de pantalla de Spotify. 2013.
13.14. Captura de pantalla de Spotify. 2013.
13.15. Captura de pantalla de Twitter. 2013.

13.16. Captura de pantalla de Twitter. 2013.
13.17. Captura de pantalla de Tumblr. 2013.
13.18. Captura de pantalla de Tumblr. 2013.

14. El mundo tableta

14.1. Touch Press. Kit de prensa. 2013.
14.2. Elaboración propia a partir de captura de pantalla de Dropbox. 2013.
14.3. Plantronicsgermany. Kit de prensa. 2013.
14.4. Elaboración propia a partir de imagen de WROBLEWSKI, Luke. 2012. http://www.lukew.com/ff/entry.asp?1649.
14.5. Spotify. Kit de prensa. 2013.
14.6. Elaboración propia a partir de captura de pantalla de Flipboard. 2013.

15. Erik Spiekermann: El Maestro de la tipografía

15.1. Elaboración propia a partir de material de Erik Spiekermann. 2013. Recuperado de: http://spiekermann.com/en/downloads/.

16. Lanzando la app

16.1. Elaboración propia a partir de captura de pantalla de Vine. 2013.
16.2. Localytics. Kit de prensa. 2013.
16.3. Simplebots. 2013. www.simplebots.co.

17. Firefox OS y Ubuntu: Los nuevos jugadores

17.1. Firefox OS. Kit de prensa. 2013
17.2. Ubuntu. Kit de prensa. 2013.

Agradecimientos

Queremos agradecer a todas las personas que de alguna forma u otra se involucraron en nuestro proyecto. Su tiempo, motivación e ideas han hecho que el resultado sea mucho mejor de lo que podríamos haber logrado sin ellas.

(Por apartados y ordenados alfabéticamente)

Entrevistas

- Patryk Adamczyk *(Mozilla)*
- Dustin Barker *(Simple)*
- Loren Brichter *(Atebits)*
- Dustin Mierau *(Path)*
- Irene Pereyra *(Fi)*
- Erik Spiekermann *(Edenspiekermann)*
- Ivo Weevers *(Ubuntu)*

Comentarios y aportes a los contenidos

- Magalí Amalla

- Daniel Armengol
- Paloma Celaá
- Pierluigi Cifani
- Andrés Colmenares
- Ruymán Ferreyra
- Armando Fidalgo
- Victoria Gerchinhoren
- Xavi Pinyol
- Florencia Rosenfeld
- Marc Torrent
- Sergi Vélez

Otras contribuciones

- Jordi Aguilà
- Jonatan Castro
- Catalina Duque Giraldo
- Rodrigo Encinas
- Esteban Humet
- Paula Marques
- Catalina Pérez
- María Daniela Quirós
- Jure Sustercic
- Seba y Maga

También queremos agradecer a nuestras familias, a Lu por el aguante y a todos los han estado cerca nuestro en estos meses, apoyándonos y difundiendo el proyecto.

Gracias totales.